

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу**

До захисту допущено:

В.о. завідувача кафедри

_____ О.Л. ТИМОЩУК

«___» _____ 20__ р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системний аналіз і управління»

спеціальності 124 «Системний аналіз»

на тему: «Групова диференціальна гра переслідування»

Виконав:

студент IV курсу, групи КА-63

Гирявець Дмитро Миколайович

Керівник:

кандидат фіз.-мат. наук, доцент

Барановська Леся Валеріївна

Консультант з нормоконтролю:

доцент, к.т.н.

Коваленко Анатолій Єпіфанович

Рецензент:

доцент кафедри СП, к.т.н., доцент

Харченко Костянтин Васильович

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 124 «Системний аналіз»

Освітньо-професійна програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ

В.о.завідувача кафедри

_____ О.Л. ТИМОЩУК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Гирявцю Дмитру Миколайовичу

1. Тема роботи «Групова диференціальна гра переслідування», керівник роботи Барановська Леся Валеріївна, кандидат ф.-м. наук, доцент, затверджені наказом по університету від «25» травня 2020 р. № 1143-с
2. Термін подання студентом роботи 08 травня 2020 року _____
3. Вихідні дані до роботи
4. Зміст роботи
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Шевчук О.А., к.е.н., доцент		

7. Дата видачі завдання 13.04.2020

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	13.04.2020	Виконано
2	Збір інформації	27.04.2020	Виконано
3	Ознайомлення з літературою і підготовка теоретичної частини диплому	27.04.2020	Виконано
4	Аналіз вимог завдання, вибір методів і засобів розв'язання поставленої задачі	04.05.2020	Виконано
5	Розробка програмного продукту	11.05.2020	Виконано
6	Виконання функціонально-вартісного аналізу	28.05.2020	Виконано
7	Отримання висновків після тестування	29.05.2020	Виконано
8	Оформлення дипломної роботи	08.06.2020	Виконано
9	Отримання допуску до захисту та подача роботи до ДЕК	12.06.2020	Виконано

Студент

Дмитро ГИРЯВЕЦЬ

Керівник

Леся БАРАНОВСЬКА

РЕФЕРАТ

Дипломна робота: 49 с., 13 рис., 6 табл., 2 дод., 11 джерел.

ГРУПОВА ДИФЕРЕНЦІАЛЬНА ГРА ПЕРЕСЛІДУВАННЯ, МЕТОД РОЗВ'ЯЗУЮЧИХ ФУНКЦІЙ

Об'єкт дослідження – групова диференціальна гра переслідування

Мета роботи – побудувати керування за допомогою методу розв'язуючих функцій у груповій диференціальній грі переслідування.

Методи дослідження – побудова алгоритмів керування.

Був досліджений метод розв'язуючих функцій для певного класу задач групового переслідування.

Для представлення траєкторій рухів переслідувачів та втікача був створений програмний продукт.

Подальший розвиток предмету дослідження – розгляд більш складних класів задач та побудова ефективного керування для втікача.

ABSTRACT

Diploma work: 49 p., 13 fig., 6 tables, 2 appendixes, 11 sources.

GROUP DIFFERENTIAL PURSUIT GAME, METHOD OF SOLVING FUNCTIONS

The object of research is a group differential pursuit game.

The purpose of the work is to build control using the method of solving functions in a group differential pursuit game.

The method of research - construction of control algorithms.

The method of solving functions for a certain class of group pursuit games was reviewed.

The software product was created to represent the trajectories of the pursuers and fugitive.

The further development of the research subject - review of more complex classes of games and building effective control for the fugitive.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	8
ВСТУП	9
РОЗДІЛ 1 ДИФЕРЕНЦІАЛЬНІ ІГРИ ТА ДОПОМІЖНІ МАТЕМАТИЧНІ РЕЗУЛЬТАТИ.....	10
1.1 Поняття диференціальних ігор переслідування	10
1.2 Класифікація задач диференціальних ігор	11
1.3 Метод розв’язуючих функцій з одним переслідувачем.....	13
Висновки до розділу 1	16
РОЗДІЛ 2 ГРУПОВА ДИФЕРЕНЦІАЛЬНА ГРА ПЕРЕСЛІДУВАННЯ	17
2.1 Зближення з термінальною множиною складної структури	17
2.2 Задача переслідування при розділених керуваннях	19
2.3 Приклад задачі для простого руху	24
2.4 Приклад задачі для процесу з простими матрицями.....	26
Висновки до розділу 2	28
РОЗДІЛ 3 ПОБУДОВА ТРАЄКТОРІЇ РУХУ ГРУПОВОГО ПЕРЕСЛІДУВАННЯ.....	29
3.1 Просте переслідування	29
3.2 Візуалізація диференціальної гри на площині	30
Висновки до розділу 3	36
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ ДЛЯ ВИРІШЕННЯ ГРУПОВОЇ ДИФЕРЕНЦІАЛЬНОЇ ГРИ ПЕРЕСЛІДУВАННЯ.....	37
4.1 Постановка задачі проектування	37
4.2 Обґрунтування функцій програмного продукту	37
4.3 Обґрунтування системи параметрів.....	39
4.4 Аналіз варіантів реалізації функцій	42
4.5 Економічний аналіз варіантів розробки програми	43

4.6 Вибір кращого варіанту ПП техніко-економічного рівня	46
Висновки до розділу 4	46
ВИСНОВКИ.....	47
ПЕРЕЛІК ПОСИЛАНЬ.....	48
ДОДАТОК А ІЛЮСТРАТИВНІ МАТЕРІАЛИ ДО ДОПОВІДІ	50
ДОДАТОК Б ТЕКСТ ПРОГРАМИ.....	55

ПЕРЕЛІК СКОРОЧЕНЬ

ККП – конфліктно-керований процес

ПП – програмний продукт

S – одинична куля з центром в нулі

R^n – n -мірний евклідів простір

ВСТУП

Задача руху супроводжує живих істот протягом всього їх існування. Щоб існувати вони мали навчитися виживати в суворих реаліях Землі. У повсякденному житті та дикій природі ми можемо побачити одну з основних ігор, в яку грають всі, а саме переслідування та втеча. Живі істоти не здатні аналізувати весь об'єм інформації, який надходить до них та пам'ятати весь шлях переслідування. На сьогоднішній день комп'ютерні технології та математичний апарат справляються з обробкою даних точніше та швидше, ніж будь-яке створіння.

Напрямок у науці, який розглядає проблеми переслідування та втечі, є диференціальні ігри. Існує безліч варіацій цієї проблематики, серед них можна виділити таку задачу, як групове переслідування.

Тема даної роботи є групова диференціальна гра переслідування, для розв'язку якої використовується метод розв'язуючих функцій.

.

РОЗДІЛ 1 ДИФЕРЕНЦІАЛЬНІ ІГРИ ТА ДОПОМІЖНІ МАТЕМАТИЧНІ РЕЗУЛЬТАТИ

1.1 Поняття диференціальних ігор переслідування

Теорія конфліктно-керованих процесів розвивається досить активно на сьогодні. Термін «диференціальні ігри» було введено одним з засновників теорії диференційних ігор Р. Айзеком [1].

У 1925 році була сформульована задача переслідування в роботі Г.Штейнгауза, яку вважають однією з перших робіт в даній області.

60-ті роки XX сторіччя вважаються роками початку розвитку теорії диференціальних ігор та пов'язуються з іменами таких математиків, як М. М. Красовський, Р. Айзекс, Л. А. Петросян, Б.М. Пшеничний, У. Флемінг, Л. С. Понтрягін,. Величезний вклад в розвиток теорії ігор внесли А.О. Чикрій, В. В. Захаров, А. А. Мелікян, А. І. Субботін, А.Азімов та багато інших.

Красовський М.М. та представники його наукової школи [2-4] створили теорію позиційних ігор, схема яких описується наступним чином [5, 6, 7]. Нехай рух конфліктно-керованого вектору z описується системою вигляду

$$\dot{z} = f(z, u, v), \quad (1.1)$$

де $z \in R^n, u \in U, v \in V, U \subset R^m, V \subset R^k, f$ – непервна функція. У просторі R^n задана непорожня цільова множина M . Сформульовані дві задачі про позиційний підхід: задача про зближення з цільовою множиною M у межах заданих фазових обмежень V ; задача про ухилення вектору від множини M . Сукупність цих (протилежних) задач є диференційною грою зближення-ухилення.

Л.С. Понтрягіним було запропоновано [8, 9] опис формалізації диференціальних ігор. Нехай рух диференційно керованого об'єкта описується системою (1.1). Л.С.

Понтрягін наголошує: «Ми пов'язуємо з диференційною грою дві різні задачі: ціллю є завершення гри, тобто приведення точки z на множину M . При цьому для здійснення цієї цілі в нашому розпорядженні знаходиться керуючий параметр u , так що в кожен момент часу t ми обираємо значення $u(t)$ цього параметра, використовуючи відомі $z(s)$ та $v(s)$ на проміжку $t - \tau < s < t$. Це правила переслідування; ціллю є запобігання кінця гри, тобто запобігання потрапляння точки z на множину M . При цьому для досягнення цієї цілі в нашому розпорядженні знаходиться керуючий параметр v , так що в кожен момент часу t ми обираємо значення $v(t)$ цього параметра, використовуючи відомі $z(s)$ та $u(s)$ на проміжку $t - \tau < s < t$. Це правила ухилення.»

У роботі [10] сформульовані достатні умови розв'язності задачі переслідування в нелінійних диференційних іграх. Це привело до створення Л.С. Понтрягіним першого та другого методів розв'язку задачі переслідування для лінійних диференційних ігор [8]. Найбільш простим та досить ефективним для розв'язку конкретних задач переслідування є перший метод Л.С.Понтрягіна, який тісно пов'язаний з методом розв'язуючих функцій, який буде розглянуто далі.

1.2 Класифікація задач диференціальних ігор

Розглянемо рух керованого об'єкта, який описується системою рівнянь[11]:

$$\dot{z} = Az + \varphi(u, v), z \in R^n, u \in U, v \in V, \quad (1.2)$$

де z – фазовий вектор, який складається з геометричних координат, швидкостей та прискорень переслідувача та втікача, A – квадратна матриця порядку n , U та V – непорожні компакти з деяких кінцевовимірних просторів, вектор-функція $\varphi(u, v): U \times V \rightarrow R^n$ неперервна за сукупністю змінних – блок керування.

Частіше всього термінальна множина має циліндричний вигляд

$$M^* = M^\circ + \varepsilon S, \quad (1.3)$$

де M° – лінійний підпростір з R^n , S – шар з ортогональних доповнень до M° в просторі R^n , $\varepsilon \geq 0$. Конфліктно-керований процес (1.2), (1.3) називається квазілінійним, якщо $\varphi(u, v) = u - v$, то процес

$$\dot{z} = Az + u - v, \quad (1.4)$$

Рівняння (1.4) може бути представлено у вигляді:

$$\begin{aligned} \dot{x} &= Bx + u, x \in R^k, \\ \dot{y} &= Cy + v, y \in R^m, \end{aligned}$$

де B, C – квадратні матриці порядку $k, m, k + m = n$.

Нехай $z(0) = z^\circ$ – початковий стан процесу (1.2). Позначимо

$$\begin{aligned} \Omega_u &= \{u(\cdot): u(t) \in U, t \geq 0, u(t) \text{ – вимірна}\} \\ \Omega_v &= \{v(\cdot): v(t) \in V, t \geq 0, v(t) \text{ – вимірна}\}. \end{aligned}$$

Функцію $u(\cdot) \in \Omega_u (v(\cdot) \in \Omega_v)$, яка була вибрана переслідувачем (втікачем) з початкової позиції z° називається програмованим керуванням переслідувача(втікача).

Для задач переслідування можуть бути представлені такі цілі[11]:

- Локальна задача зближення з фіксованим(нефіксованим) часом. Знайти умови на параметри процесу(1.2) і початковий стан z° , які будуть достатні для того, щоб траєкторія процесу(1.2) була приведена на термінальну множину з початкового стану в деякий момент(не пізніше деякого моменту) $T(z^\circ)$ в заданному класі керування переслідувача при будь-яких протидіях втікача. $T(z^\circ)$ не відомий заздалегідь.
- Глобальна задача зближення (повна конфліктна керованість). Знайти умови на параметри процесу(1.2), які будуть достатні для того, щоб траєкторія процесу(1.2) була приведена на термінальну множину з будь-якого початкового стану z° за кінцевий час в заданному класі керування переслідувача при будь-яких протидіях втікача.
- Локальна задача ухилення на кінцевому (напівнескінченному) інтервалі. Знайти умови на параметри процесу(1.2) і початковий стан z° , при яких існує стратегія втікача, аби траєкторія процесу(1.2) не потрапила на термінальну множину на інтервалі часу $[0, T], ([0, +\infty))$ при будь-якому керуванні переслідувача.
- Глобальна задача ухилення. Знайти умови на параметри процесу(1.2), при яких можлива втеча від зустрічі з термінальною множиною в заданному класі керування з будь-якого положення на всьому напівнескінченному інтервалі часу при будь-якому керуванні переслідувача.

1.3 Метод розв'язуючих функцій з одним переслідувачем

Розглянемо локальну задачу зближення з фіксованим часом в класі квазістратегій, де один переслідувач та один втікач[11]:

$$\dot{z} = Az + \varphi(u, v), z \in R^n, u \in U, v \in V, \quad (1.5)$$

де A – постійна квадратна матриця порядку n , U та V – непорожні компакти з деяких кінцевовимірних просторів, вектор-функція $\varphi(u, v): U \times V \rightarrow R^n$ неперервна за сукупністю змінних.

Термінальна множина має вигляд

$$M^* = M^\circ + M, \quad (1.6)$$

де M° – лінійний підпростір з R^n , M – непорожні компакти з ортогональних доповнень L до M° в просторі R^n . Нехай π – оператор ортогонального проектування з R^n на L . Введемо багатозначне відображення

$$W(t, v) = \pi e^{At} \varphi(U, v)$$

$$W(t) = \bigcap_{v \in V} W(t, v), t \geq 0.$$

Умова 1.3.1.(умова Л.С.Понтрягіна) Відображення $W(t) \neq \emptyset, t \geq 0$.

$W(t)$ – напівнеперервне зверху, отже воно борелівське. Це означає, що існує хоча б один борелівський селектор $\gamma(t)$ відображення $W(t)$. Нехай $\Gamma = \{\gamma(\cdot): \gamma(t) \in W(t), t \geq 0\}$ сукупність борелівських селекторів багатозначного відображення $W(t)$.

Зафіксуємо деякий елемент та покладемо

$$\xi(t, z, \gamma(\cdot)) = \pi e^{At} z + \int_0^t \gamma(\tau) d\tau.$$

Розглянемо розв'язуючу функцію

$$\alpha(t, \tau, z, v, \gamma(\cdot)) = \\ = \sup \{ \alpha \geq 0 : [W(t - \tau, v) - \gamma(t - \tau)] \cap \alpha(M - \xi(t, z, \gamma(\cdot))) \neq \emptyset \}. \quad (1.7)$$

Якщо $\xi(t, z, \gamma(\cdot)) \in M$, то покладемо $\alpha(t, \tau, z, v, \gamma(\cdot)) = +\infty, 0 \leq \tau \leq t$, $v \in V$. В інших випадках $\alpha(t, \tau, z, v, \gamma(\cdot))$ обмежена при будь-яких $\tau \in [0, t]$, $v \in V$.

Лема 1.3.1. Нехай для конфліктно керованого процесу (1.5), (1.6) виконується умова 1.3.1. і для деяких

$$t, z, \gamma(\cdot), t > 0, z \in R^n, \gamma(\cdot) \in \Gamma, \xi(t, z, \gamma(\cdot)) \notin M.$$

Тоді розв'язуюча функція (1.7) борелівська по (τ, v) і напівнеперервна зверху по $v, \tau \in [0, t]$, $v \in V$.

Розглянемо функцію

$$T(z, \gamma(\cdot)) = \inf \{ t \geq 0 : \int_0^t \inf_{v(\cdot) \in V} \alpha(t, \tau, z, v(\tau), \gamma(\cdot)) d\tau \geq 1 \}. \quad (1.8)$$

Якщо нерівність не виконується, то покладемо $T(z, \gamma(\cdot)) = +\infty$.

Лема 1.3.2 Нехай для конфліктно керованого процесу (1.5), (1.6) виконується умова 1.3.1. і для деяких, $z \in R^n, \gamma(\cdot) \in \Gamma, T(z, \gamma(\cdot)) < +\infty$. Тоді нижня границя у виразі(1.8) досягається[11].

Теорема 1.3.1 [Чикрій]. Нехай для конфліктно керованого процесу (1.5),(1.6), виконується умова 1.3.1, M – випукла множина, z° - початковий стан процесу (1.5) б і для деякого борелівський селектор $\gamma^\circ(\cdot) \in \Gamma$ виконується нерівність $T(z^\circ, \gamma^\circ(\cdot)) < +\infty$. Тоді траєкторія процесу (1.5) може бути приведена на термінальну множину в момент $T_\vartheta(z^\circ, \gamma^\circ(\cdot))$.

Лема 1.3.3 Нехай конфліктно керований процесу (1.5),(1.6) лінійний, виконується умова 1.3.1, існує неперервна додатня числова функція $r(t), r: R^+ \rightarrow R^+$, і число $l > 0$, такі, що $\pi e^{At}U = r(t)S, M = lS$, де S - куля одиничного радіусу з центром у нулі підпростору L . Тоді розв'язуюча функція

$$\alpha(t, \tau, z, v, \gamma(\cdot)), 0 \leq \tau \leq t, z \in R^n, v \in V, \gamma(\cdot) \in \Gamma,$$

при $\xi(t, z, \gamma(\cdot)) \notin M$ є більшим додатнім корнем квадратного рівняння

$$\|\pi e^{A(t-\tau)}v + \gamma(t - \tau) - \alpha\xi(t, z, \gamma(\cdot))\| = r(t - \tau) + \alpha l$$

відносно α [11].

Висновки до розділу 1

Отже, розв'язок групової задачі переслідування є актуальною темою для розвитку. Він оснований на базі конфліктно-керованих процесів, які описуються системою диференціальних рівнянь.

В першому розділі ми розглянули основні поняття диференціальних ігор, поняття та теореми, які застосовуються у знаходженні оптимального керування. Також розглянули випадок знаходження оптимального керування для випадку одного переслідувача та втікача.

РОЗДІЛ 2 ГРУПОВА ДИФЕРЕНЦІАЛЬНА ГРА ПЕРЕСЛІДУВАННЯ

2.1 Зближення з термінальною множиною складної структури

Розглянемо локальну задачу зближення з фіксованим часом в класі квазістратегій:

$$\dot{z} = Az + \varphi(u, v), z \in R^n, u \in U, v \in V, \quad (2.1)$$

де A – квадратна матриця порядку n , U та V – непорожні компакти з деяких скінченновимірних просторів, вектор-функція $\varphi(u, v): U \times V \rightarrow R^n$ неперервна за сукупністю змінних.

Термінальна множина має вигляд

$$M^* = \bigcup_{l=1, \dots, \vartheta} \{M_l^\perp + M_l\}, \quad (2.2)$$

де M_l^\perp – лінійний підпростір з R^n , M_l – непорожні компакти з ортогональних доповнень L_l до M_l^\perp в просторі R^n .

Нехай π_l – оператор ортогонального проектування з R^n на L_l . Введемо багатозначне відображення

$$F(t) = \bigcap_{v \in V} \bigcup_{u \in U} (\pi_1 e^{At} \varphi(u, v), \dots, \pi_\vartheta e^{At} \varphi(u, v)), t \geq 0$$

Умова 2.1 $\text{dom } F(t) = [0; \infty)$.

Виходячи з припущення про параметри процесу (2.1), (2.2) та умови 2.1, $F(t)$ напівнеперервне зверху, отже вона борелівське. Це означає, що існує хоча

б один борелівський селектор $\gamma(t) = \text{column}(\gamma_1(t), \dots, \gamma_{\vartheta}(t))$ відображення $F(t)$. Зафіксуємо його та покладемо

$$\xi_l(t, z, \gamma_l(\cdot)) = \pi_l e^{At} z + \int_0^t \gamma_l(\tau) d\tau, \quad l = \overline{1, \vartheta}.$$

Нехай

$$R_{\vartheta}^+ = \{\alpha = (\alpha_1, \dots, \alpha_{\vartheta}) : \alpha_l \geq 0, \quad l = \overline{1, \vartheta}, t \geq \tau \geq 0, v \in V, z \in R^n.$$

Визначимо багатозначне відображення

$$\begin{aligned} \Sigma(t, \tau, z, v, \gamma(\cdot)) &= \{\alpha \in R_{\vartheta}^+ : \\ &\left[\bigcup_{u \in U} \pi_1 e^{A(t-\tau)} \varphi(u, v) - \gamma_1(t-\tau), \dots, \pi_{\vartheta} e^{A(t-\tau)} \varphi(u, v) - \gamma_{\vartheta}(t-\tau) \right] \\ &\bigcap [\alpha_1 (M_1 - \xi_1(t, z, \gamma_1(\cdot))), \dots, \alpha_{\vartheta} (M_{\vartheta} - \xi_{\vartheta}(t, z, \gamma_{\vartheta}(\cdot)))] \neq \emptyset\}. \end{aligned}$$

Якщо $\xi_l(t, z, \gamma_l(\cdot)) \notin M_l$ для всіх $l = \overline{1, \vartheta}$ та для деякого $t \geq 0$, то $\Sigma(t, \tau, z, v, \gamma(\cdot))$ обмежене та борелівське по τ та по v . Виберемо селектор

$$\alpha(t, \tau, z, v, \gamma(\cdot)) = \text{lexmax } \Sigma(t, \tau, z, v, \gamma(\cdot)), \quad (2.3)$$

який є борелівським по τ та по v . Якщо для деякого моменту $t \geq 0$ і для l : $\xi_l(t, z, \gamma_l(\cdot)) \in M_l$, то покладемо $\alpha_l(t, \tau, z, v, \gamma_l(\cdot)) = +\infty$, інші обмежені компоненти виберемо як у виразі (2.3). Покладемо

$$T_{\vartheta}(z, \gamma(\cdot)) = \min \left\{ t \geq 0 : \inf_{v(\cdot) \in \Omega_V} \max_{l=1, \dots, \vartheta} \int_0^t \alpha_l(t, \tau, z, v(\tau), \gamma_l(\cdot)) d\tau \geq 1 \right\}.$$

Якщо нерівність не виконується, то покладемо $T_\vartheta(z, \gamma(\cdot)) = +\infty$.

Теорема 2.1 [Чикрій]. Нехай для конфліктно керованого процесу (2.1), (2.2) виконується умова 2.1, множини $M_l, l = \overline{1, \vartheta}$ випуклі і існує борелівський селектор $\gamma^\circ(t)$ багатозначного відображення $F(t)$, що $T_\vartheta(z^\circ, \gamma^\circ(\cdot)) < +\infty$, де z° - початковий стан процесу (2.1). Тоді траєкторія процесу (2.1) може бути приведена на термінальну множину M^* з початкового стану z° в момент $T_\vartheta(z^\circ, \gamma^\circ(\cdot))$.

2.2 Задача переслідування при розділених керуваннях

Розглянемо локальну задачу зближення з фіксованим часом, яка задається системою диференціальних рівнянь:

$$\dot{z}_l = A_l z_l + \varphi_l(u_l, v), z_l \in R^{n_l}, l = \overline{1, \vartheta}, u_l \in U_l, v \in V, \quad (2.4)$$

де A_l – квадратна матриця порядку $n_l, n_1 + \dots + n_\vartheta = n$, U_l та V – непорожні компакти з деяких скінченновимірних просторів, вектор-функція $\varphi_l(u_l, v): U_l \times V \rightarrow R^{n_l}$ неперервна за сукупністю змінних.

Термінальна множина має вигляд

$$M^* = \bigcup_{l=1, \dots, \vartheta} \{M_l^\circ + M_l\} = \bigcup_{l=1, \dots, \vartheta} M_l^*, \quad (2.5)$$

де M_l° - лінійний підпростір з R^{n_l} , M_l – випуклий компакт з ортогональних доповнень L_l до M_l° в просторі R^{n_l} .

Гра (2.4), (2.5) вважається закінченою, якщо для деякого $l = \overline{1, \vartheta}$ виконується умова $z_l \in M_l$. Переслідувачі використовують квазістратегії, а втікач - програмне керування.

Нехай π_l – оператор ортогонального проектування з R^{n_l} на L_l .
Розглянемо багатозначні відображення

$$W_l(t, v) = \pi_l e^{A_l t} \varphi_l(U_l, v)$$

$$W_l(t) = \bigcap_{v \in V} W_l(t, v), t \geq 0, v \in V.$$

Умова 2.2 Відображення $W_l(t) \neq \emptyset$ для всіх $l = \overline{1, \vartheta}, t \geq 0$.

$W_l(t)$ - напівнеперервне зверху, виходячи з умови 2.2, отже воно борелівське. Це означає, що існує хоча б один борелівський селектор $\gamma_l(t)$ відображення $W_l(t)$. Зафіксуємо його та покладемо

$$\xi_l(t, z_l, \gamma_l(\cdot)) = \pi_l e^{A_l t} z_l + \int_0^t \gamma_l(\tau) d\tau.$$

Кожному переслідуювачу поставимо у відповідність розв'язуючу функцію

$$\alpha_l(t, \tau, z_l, v, \gamma_l(\cdot)) =$$

$$= \sup \{ \alpha_l \geq 0 : [W_l(t - \tau, v) - \gamma_l(t - \tau)] \bigcap \alpha_l(M_l - \xi_l(t, z_l, \gamma_l(\cdot))) \neq \emptyset \}$$

Якщо $\xi_l(t, z, \gamma_l(\cdot)) \in M_l$, то покладемо $\alpha_l(t, \tau, z, v, \gamma_l(\cdot)) = +\infty, 0 \leq \tau \leq t, v \in V$. В інших випадках $\alpha_l(t, \tau, z, v, \gamma_l(\cdot))$ обмежена при будь-яких $\tau \in [0, t], v \in V$. Візьмемо $\gamma(\cdot) = column(\gamma_1(\cdot), \dots, \gamma_\vartheta(\cdot))$ і нехай

$$\Gamma_\vartheta = \{ \gamma(\cdot) : \gamma_l(t) \in W_l(t), t \geq 0, l = \overline{1, \vartheta} \}$$

$$T_\vartheta(z, \gamma(\cdot)) = \min \left\{ t \geq 0 : \inf_{v(\cdot) \in \Omega_V} \max_{l=1, \dots, \vartheta} \int_0^t \alpha_l(t, \tau, z_l, v(\tau), \gamma_l(\cdot)) d\tau \geq 1 \right\}.$$

Якщо нерівність не виконується, то покладемо $T_{\vartheta}(z, \gamma(\cdot)) = +\infty$. Інакше мінімум досягається (лема 1.3.2).

Теорема 2.2 [Чикрій]. Нехай для конфліктно керованого процесу (2.4), (2.5), виконується умова 2.2, z° - початковий стан процесу (2.4) б і для деякого борелівський селектор $\gamma^{\circ}(t) \in \Gamma_{\vartheta}$ виконується нерівність $T_{\vartheta}(z^{\circ}, \gamma^{\circ}(\cdot)) < +\infty$. Тоді траєкторія процесу (2.4) хоча б одного l може бути приведена на термінальну множину M_l^* з початкового стану z° в момент $T_{\vartheta}(z^{\circ}, \gamma^{\circ}(\cdot))$.

Покладемо $T = T_{\vartheta}(z^{\circ}, \gamma^{\circ}(\cdot))$ і нехай $v(\tau) \in V, 0 \leq \tau \leq T$ - довільна вимірна функція. Розглянемо випадок, коли $\xi_l(T, z_l^{\circ}, \gamma_l^{\circ}(\cdot)) \notin M_l$ для всіх $l = \overline{1, \vartheta}$. Тоді контрольна функція

$$h_{\vartheta}(t) = h_{\vartheta}(T, t, z^{\circ}, v(\cdot), \gamma(\cdot)) = 1 - \max_{l=1, \dots, \vartheta} \int_0^t \alpha_l(T, \tau, z^{\circ}, v(\tau), \gamma_l^{\circ}(\cdot)) d\tau$$

є неперервною та монотонно спадною по t . Визначимо момент переключення $t_* = t_*(v(\cdot)), t_* \in (0, T)$ з рівності

$$h_{\vartheta}(t_*) = 0. \quad (2.6)$$

Розглянемо багатозначне відображення

$$U_1^l(\tau, v) = \{u_l \in U_l : \pi_l e^{A_l(T-\tau)} \varphi_l(u_l, v) - \gamma_l(T-\tau) \in \alpha_l(T, \tau, z^{\circ}, v(\tau), \gamma_l^{\circ}(\cdot)) (M_l - \xi_l(T, z_l^{\circ}, \gamma_l^{\circ}(\cdot)))\} l = \overline{1, \vartheta}. \quad (2.7)$$

Вони є борелівськими по (τ, v) . Тоді селектори

$$u_1^l(\tau, v) = \text{lexmin } U_1^l(\tau, v), l = \overline{1, \vartheta} \quad (2.8)$$

є борелівськими у сукупності (τ, v) функціями.

Керування переслідувачів на інтервалі $[0, t_*]$ покладемо рівними

$$u_l(\tau) = u_1^l(\tau, v(\tau)), l = \overline{1, \vartheta}. \quad (2.9)$$

З рівності (2.6) отримуємо, що існує номер l_* , такий, що

$$1 - \int_0^t \alpha_{l_*} \left(T, \tau, z_{l_*}^\circ, v(\tau), \gamma_{l_*}^\circ(\cdot) \right) d\tau = 0. \quad (2.10)$$

Розглянемо багатозначне відображення

$$U_2^l(\tau, v) = \{u_l \in U_l : \pi_l e^{A_l(T-\tau)} \varphi_l(u_l, v) - \gamma_l^\circ(T - \tau) = 0\}, \quad (2.11)$$

яке є борелівським по (τ, v) . Тоді селектор

$$u_2^l(\tau, v) = \text{lexmin } U_2^l(\tau, v) \quad (2.12)$$

є борелівською за (τ, v) функцією.

На інтервалі $(t_*, T]$ покладемо $\alpha_{l_*} \left(T, \tau, z_{l_*}^\circ, v, \gamma_{l_*}^\circ(\cdot) \right) = 0$, а керування переслідувача l_* -

$$u_{l_*}(\tau) = u_2^{l_*}(\tau, v(\tau)), \quad (2.13)$$

а керування інших переслідувачів на інтервалі виберемо довільними. Якщо існує такий номер l , що $\xi_l \left(T, z_l^\circ, \gamma_l^\circ(\cdot) \right) \in M_l$, то керування l -го переслідувача покладемо рівним

$$u_l(\tau) = u_2^l(\tau, v(\tau)), 0 \leq \tau \leq T,$$

а керування інших переслідувачів довільні. Розглянемо випадок $\xi_l(T, z_l^\circ, \gamma_l^\circ(\cdot)) \notin M_l \forall l = \overline{1, \vartheta}$ та прослідкуємо за переслідувачем l_* .

З рівності (2.10) та $\alpha_{l_*}(T, \tau, z_{l_*}^\circ, v, \gamma_{l_*}^\circ(\cdot)) = 0, \tau \in (t_*, T], v(\tau) \in V$, отримуємо

$$\int_0^T \alpha_{l_*}(T, \tau, z_{l_*}^\circ, v(\tau), \gamma_{l_*}^\circ(\cdot)) d\tau = 1. \quad (2.14)$$

З формули Коші для l_* -го переслідувача маємо

$$\pi_{l_*} z_{l_*}(T) = \pi_{l_*} e^{A_{l_*} T} z_{l_*}^\circ + \int_0^T \pi_{l_*} e^{A_{l_*}(T-\tau)} \varphi_{l_*}(u_{l_*}(\tau), v(\tau)) d\tau.$$

При додаванні та відніманні в првій частині величину $\int_0^T \gamma_{l_*}^\circ(T-\tau) d\tau$, і враховуючи закони вибору керування переслідувачів (2.7) – (2.9), (2.11) – (2.13), отримаємо включення

$$\begin{aligned} \pi_{l_*} z_{l_*}(T) \in \xi_{l_*}(T, z_{l_*}^\circ, \gamma_{l_*}^\circ(\cdot)) & \left(1 - \int_0^T \alpha_{l_*}(T, \tau, z_{l_*}^\circ, v(\tau), \gamma_{l_*}^\circ(\cdot)) d\tau \right) + \\ & + \int_0^T \alpha_{l_*}(T, \tau, z_{l_*}^\circ, v(\tau), \gamma_{l_*}^\circ(\cdot)) M_{l_*} d\tau. \end{aligned}$$

Звідси з урахуванням (2.14) і леми 1.2.10 випливає включення $\pi_{l_*} z_{l_*}(T) \in M_{l_*}$. Якщо для деякого l $\xi_l(T, z_l^\circ, \gamma_l^\circ(\cdot)) \in M_l$, то, враховуючи закон вибору керування в цьому випадку, вираз для $\xi_l(T, z_l^\circ, \gamma_l^\circ(\cdot))$ і формулу Коші, отримаємо включення $\pi_l z_l(T) \in M_l$.

Наслідок 2.1 [Чикрій]. Нехай для конфліктно керованого процесу (2.4), (2.5), виконується умова 2.2, $\varphi_l(u_l, v) = u_l - v$, багатозначні відображення $\pi_l e^{A_l t} U_l = r_l(t) S_l$, множини $M_l = \epsilon_l S_l, l = \overline{1, \vartheta}$, де $r_l(t)$ – неперервні невід’ємні числові функції, ϵ_l – невід’ємні числа, S_l – куля одиничного радіусу з центром у нулі підпростору L_l . Тоді розв’язуючі функції $\alpha_l(t, \tau, z_l, v(\tau), \gamma_l(\cdot))$ при $\xi_l(t, z_l, \gamma_l(\cdot)) \notin M_l$ є найбільшими коренями відповідних квадратних рівнянь

$$\begin{aligned} \|\pi_l e^{A_l(t-\tau)} v + \gamma_l(t - \tau) - \alpha_l \xi_l(t, z_l, \gamma_l(\cdot))\| &= r_l(t - \tau) + \alpha_l \epsilon_l, l = \overline{1, \vartheta}, \\ 0 \leq \tau \leq t, v \in V, \gamma_l(t) &\in W_l(t) \end{aligned}$$

відносно $\alpha_l, \alpha_l \geq 0$.

2.3 Приклад задачі для простого руху

Розглянемо задачу групового переслідування з ϑ переслідувачами та одним втікачем[11]

$$\dot{x}_l = u_l, \quad x_l \in R^k, k \geq 1, \|u_l\| \leq 1, l = \overline{1, \vartheta} \quad (2.15)$$

$$\dot{y} = v, \quad y \in R^k, \|v\| \leq 1. \quad (2.16)$$

Гра вважається закінченою, якщо для деякого l $x_l = y$. Тоді з (2.15), (2.16) отримаємо завдяки заміні $z_l = x_l - y$ такий процес

$$\dot{z}_l = u_l - v, \quad z_l \in R^k, \|u_l\| \leq 1, \|v\| \leq 1, l = \overline{1, \vartheta}. \quad (2.17)$$

З термінальною множиною

$$M^* = \bigcup_{l=1,\dots,\vartheta} \{M_l^\circ\} = \bigcup_{l=1,\dots,\vartheta} \{z_l : z_l = 0\}.$$

Отримуємо $W_l(t) = \{0\}, l = \overline{1, \vartheta}$, отже селектори $\gamma_l(t) = 0$ функції $\xi_l(t, z_l, \gamma_l(\cdot)) = z_l, l = \overline{1, \vartheta}$, виходить, що розв'язуючі функції переслідувачів $z_l \neq 0$

$$\alpha_l(t, \tau, z_l, v, 0) = \alpha_l(z_l, v) = \max\{\alpha_l \geq 0 : -\alpha_l z_l \in S - v\}.$$

З наслідку 2.1. випливає, що вони мають вигляд

$$\alpha_l(z_l, v) = \frac{(z_l, v) + ((z_l, v)^2 + \|z_l\|^2(1 - \|v\|^2))^{1/2}}{\|z_l\|^2}, l = \overline{1, \vartheta}. \quad (2.18)$$

Час завершення групового переслідування буде

$$T_\vartheta(z, 0) = T_\vartheta(z) = \min \left\{ t \geq 0 : \inf_{v(\cdot) \in \Omega_V} \max_{l=1,\dots,\vartheta} \int_0^t \alpha_l(z_l, v(\tau)) d\tau = 1 \right\}.$$

Позначимо $\delta(z) = \min_{\|v\| \leq 1} \max_{l=1,\dots,\vartheta} \alpha_l(z_l, v)$, і маємо обмеження зверху[13]

$$T_\vartheta(z) \leq \frac{\vartheta}{\delta(z)}.$$

Теорема 2.3 [Чикрій]. Нехай z° - початковий стан (2.17). Тоді, якщо $0 \in \text{int co}\{\frac{z_l^\circ}{\|z_l^\circ\|}\}$, то задача групового переслідування буде завершена в момент

$T_{\vartheta}(z^{\circ})$, для якого справедлива оцінка $T_{\vartheta}(z) \leq \frac{\vartheta}{\delta(z)}$. При цьому, якщо $t_* = t_*(v(\cdot))$ – момент переключення, який є нулем контрольної функції $1 - \int_0^t \alpha_l(z_l^{\circ}, v(\tau)) d\tau$, то керування переслідувачів, які реалізують час $T_{\vartheta}(z^{\circ})$, на інтервалі $[0, t_*]$ має вигляд

$$u_l(\tau) = v(\tau) - \alpha_l(z_l^{\circ}, v(\tau)) z_l^{\circ}, l = \overline{1, \vartheta},$$

А на інтервалі $(t_*, T_{\vartheta}(z^{\circ})]$ - $u_l(\tau) = v(\tau)$ для індексів l , які задовільняють рівності

$$\int_0^t \alpha_l(z_l^{\circ}, v(\tau)) d\tau = \max_{l=1, \dots, \vartheta} \int_0^t \alpha_l(z_l^{\circ}, v(\tau)) d\tau$$

а для інших індексів керування довільні на інтервалі $(t_*, T_{\vartheta}(z^{\circ})]$. $0 \notin \text{int co}\{\frac{z_l^{\circ}}{\|z_l^{\circ}\|}\}$, то вирішувана задача втечі.

2.4 Приклад задачі для процесу з простими матрицями

Розглянемо задачу групового переслідування з ϑ переслідувачами та одним втікачем[11]:

$$\dot{z}_l = a_l z_l + u_l - v, a_l < 0, z_l \in R^k, \|u_l\| \leq 1, \|v\| \leq 1, l = \overline{1, \vartheta} \quad (2.19)$$

Гра вважається закінченою, якщо для деякого l : $x_l = y$.

Термінальна множина має вигляд

$$M^* = \bigcup_{l=1,\dots,\vartheta} \{M_l^\circ\} = \bigcup_{l=1,\dots,\vartheta} \{z_l : z_l = 0\}.$$

Отримуємо $W_l(t) = \{0\}, l = \overline{1, \vartheta}$, отже селектори $\gamma_l(t) = 0$ функції $\xi_l(t, z_l, \gamma_l(\cdot)) = e^{a_l t} z_l, l = \overline{1, \vartheta}$, виходить, що розв'язуючі функції переслідувачів $z_l \neq 0$

$$\alpha_l(t, \tau, z_l, v, 0) = \max\{\alpha_l \geq 0 : -\alpha_l e^{a_l t} z_l \in e^{a_l(t-\tau)}(S - v)\} = e^{-a_l \tau} \alpha_l(z_l, v)$$

де $\alpha_l(z_l, v)$ має вигляд (2.18).

Час завершення групового переслідування

$$T_\vartheta(z) = \min \left\{ t \geq 0 : \inf_{v(\cdot) \in \Omega_V} \max_{l=1,\dots,\vartheta} \int_0^t e^{-a_l \tau} \alpha_l(z_l, v(\tau)) d\tau = 1 \right\}.$$

Позначимо $\delta(z) = \min_{||v|| \leq 1} \max_{l=1,\dots,\vartheta} \alpha_l(z_l, v)$, отримали обмеження зверху

$$T_\vartheta(z) \leq \frac{1}{a_*} \ln \left(1 + \frac{a_* \vartheta}{\delta(z)} \right) \quad (2.20)$$

де $a_* = -\max_{l=1,\dots,\vartheta} a_l$ [11].

Теорема 2.4 [Чикрій]. Нехай z° - початковий стан (2.19). Тоді, якщо $0 \in \text{int } co\{z_l^\circ\}$, то задача групового переслідування вирішувана в момент $T_\vartheta(z^\circ)$, для якого справедлива оцінка (2.20). При цьому, якщо $t_* = t_*(v(\cdot))$ - момент переключення $t_* \leq T_\vartheta(z^\circ)$, який є нулем контрольної функції

$$1 - \max_{l=1,\dots,\vartheta} \int_0^t e^{-a_l \tau} \alpha_l(z_l, v(\tau)) d\tau,$$

то керування переслідувачів, які реалізують час $T_{\vartheta}(z^{\circ})$, на інтервалі $[0, t_*]$ має вигляд

$$u_l(\tau) = v(\tau) - \alpha_l(z_l^{\circ}, v(\tau)) z_l^{\circ}, l = \overline{1, \vartheta},$$

а на інтервалі $(t_*, T_{\vartheta}(z^{\circ})]$ - $u_l(\tau) = v(\tau)$ для індексів l , які задовільняють рівності

$$\int_0^t e^{-a_l \tau} \alpha_l(z_l^{\circ}, v(\tau)) d\tau = \max_{l=1, \dots, \vartheta} \int_0^t e^{-a_l \tau} \alpha_l(z_l^{\circ}, v(\tau)) d\tau,$$

для інших індексів керування довільні на інтервалі $(t_*, T_{\vartheta}(z^{\circ})]$.

Якщо $0 \notin \text{int co}\{z_l^{\circ}\}$, то вирішувана задача втечі, яка реалізується на будь-якому постійному керуванні втікача, на якому досягається мінімум функції $\max_{l=1, \dots, \vartheta} \alpha_l(z^{\circ}, v)$.

Висновки до розділу 2

У даному розділі був розглянутий метод розв'язуючих функцій для групової диференціальної гри переслідування. Були розглянуті теореми та основні поняття в постановці задачі групового переслідування.

Показали застосування методу на декількох прикладах та теореми, в яких сформульовані необхідні умови завершення гри переслідування та побудовані відповідні керування.

РОЗДІЛ 3 ПОБУДОВА ТРАЄКТОРІЇ РУХУ ГРУПОВОГО ПЕРЕСЛІДУВАННЯ

3.1 Просте переслідування

Розглянемо просту модель задачі групового переслідування. Нехай є ϑ переслідувачів та один втікач. Вважаємо, що положення учасників є геометричні точки, тобто не враховуємо їх розміри. Закони рухів переслідувачів мають вигляд

$$\dot{x}_l = u_l, \quad x_l \in R^k, k \geq 1, ||u_l|| \leq a_l, l = \overline{1, \vartheta}.$$

Закон руху втікача

$$\dot{y} = v, \quad y \in R^k, ||v|| \leq b.$$

Отже, гравці рухаються з обмеженими швидкостями. Числа a_l та b показують максимальні швидкості гравців. Вважаємо гру закінченою якщо для деякого l $x_l = y$.

Нехай $a_l \geq b, l = \overline{1, \vartheta}$ та існує певна дискретизація часу. Вважаємо, що гравці вибирають свої керування в кожний момент дискретизації та пересуваються на вказані ними напрямки, враховуючи їх швидкісні обмеження. Втікач знає про положення переслідувачів в даний момент, а переслідувачі знають положення втікача та керування, яке він вибрав в даний момент. Застосуємо теорію та алгоритм пошуку керування переслідувачів викладену в попередньому розділі. Отже, керування визначається наступним алгоритмом:

1. Рахуємо $\alpha_l(z_l, v)$ за формулою

$$\alpha_l(z_l, v) = \frac{(z_l, v) + ((z_l, v)^2 + \|z_l\|^2 (a_l^2 - \|v\|^2))^{1/2}}{\|z_l\|^2}, l = \overline{1, \vartheta}.$$

2. Перевіряємо, чи $\alpha_l(z_l, v) < 1, l = \overline{1, \vartheta}$. Якщо так, то керування рахуємо за формулами $u_l = v - \alpha_l(z_l, v)z_l, l = \overline{1, \vartheta}$.

3. Якщо для певних l : $\alpha_l(z_l, v) \geq 1$, то з них знаходимо переслідувача з найбільшим значенням l_{max} : $\alpha_{l_{max}} = \max \alpha_l(z_l, v)$ та визначаємо його керування $u_{l_{max}} = v - z_{l_{max}}$, а керування всіх інших покладаємо нулю.

3.2 Візуалізація диференціальної гри на площині

Одним з найважливіших етапів створення ПП є проектування інтерфейсу. Через інтерфейс відбувається взаємодія між користувачем та програмою, і від нього залежить, які рішення буде приймати користувач.

Для представлення траєкторій рухів переслідувачів та втікача у грі, описаній у попередньому пункті та для випадку площини, був створений ПП на мові програмування Python. ПП є прототипом системи моделювання «переслідувачі-втікач», яку можна буде застосовувати для вибору керування в задачах керування.

Для програми вхідними даними є наступні відомості:

- кількість переслідувачів;
- початкові координати всіх учасників;
- максимальні значення швидкості для всіх учасників.

Для того, щоб алгоритм з попереднього пункту працював потрібен вибір керування втікача. В розробці було розглянуто два випадки вибору керування втікача. В першому, керування втікача будується за наступним алгоритмом:

- 1) втікач визначає найближчого переслідувача в плані евклідової

норми;

2) втікач будує своє керування на промені, який виходить з положення найближчого переслідувача та перетинає положення втікача, у напрямку протилежному переслідувачу з максимальною швидкістю.

В другому випадку, керування втікача задає користувач.

На графіках початкові точки переслідувачів відмічені зірками, початкове положення втікача - ромбом, а точка перехоплення – п'ятикутником.

Розглянемо спочатку більш детально перший випадок. Для нього характерним є те, що втікач не оптимально вибирає своє керування, а зважає лише на найближчого переслідувача. Подібні дії можна побачити у реальному житті, коли з всіх загроз, об'єкт помічає лише найближчу та не звертає увагу на інші.

На рис. 3.1 - рис. 3.5 показані декілька прикладів при різній кількості переслідувачів та обмеженні швидкості.

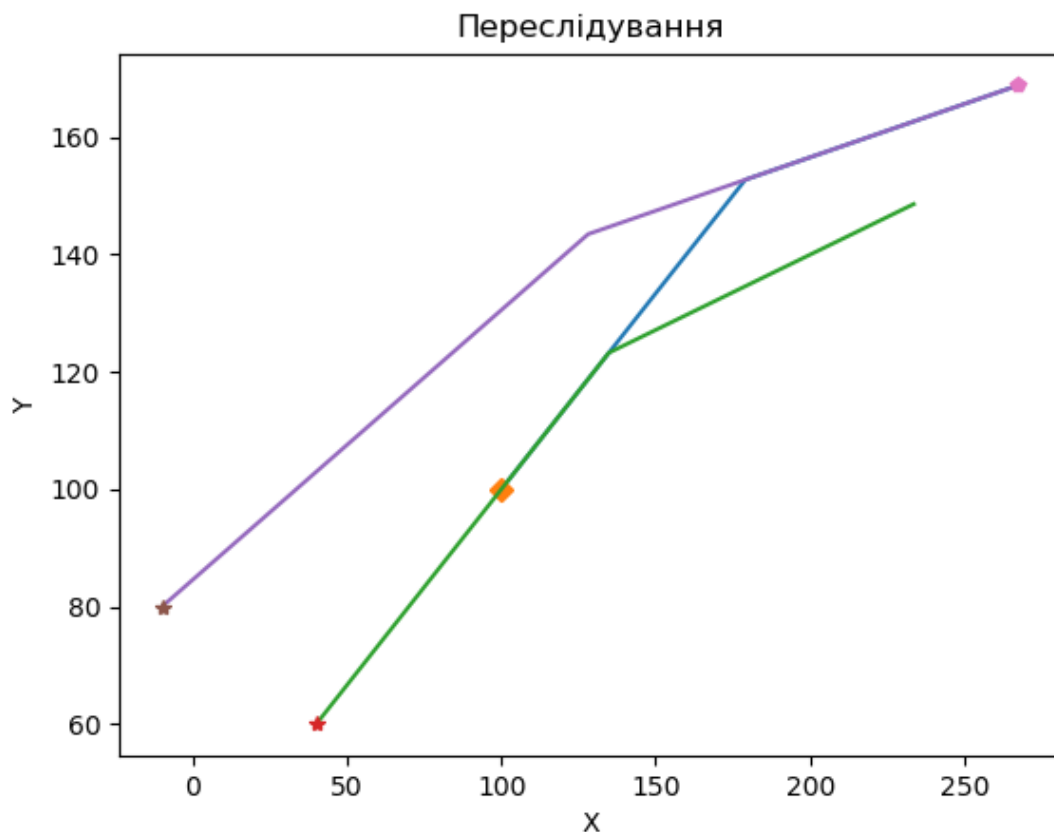


Рисунок 3.1 – 2 переслідувачі

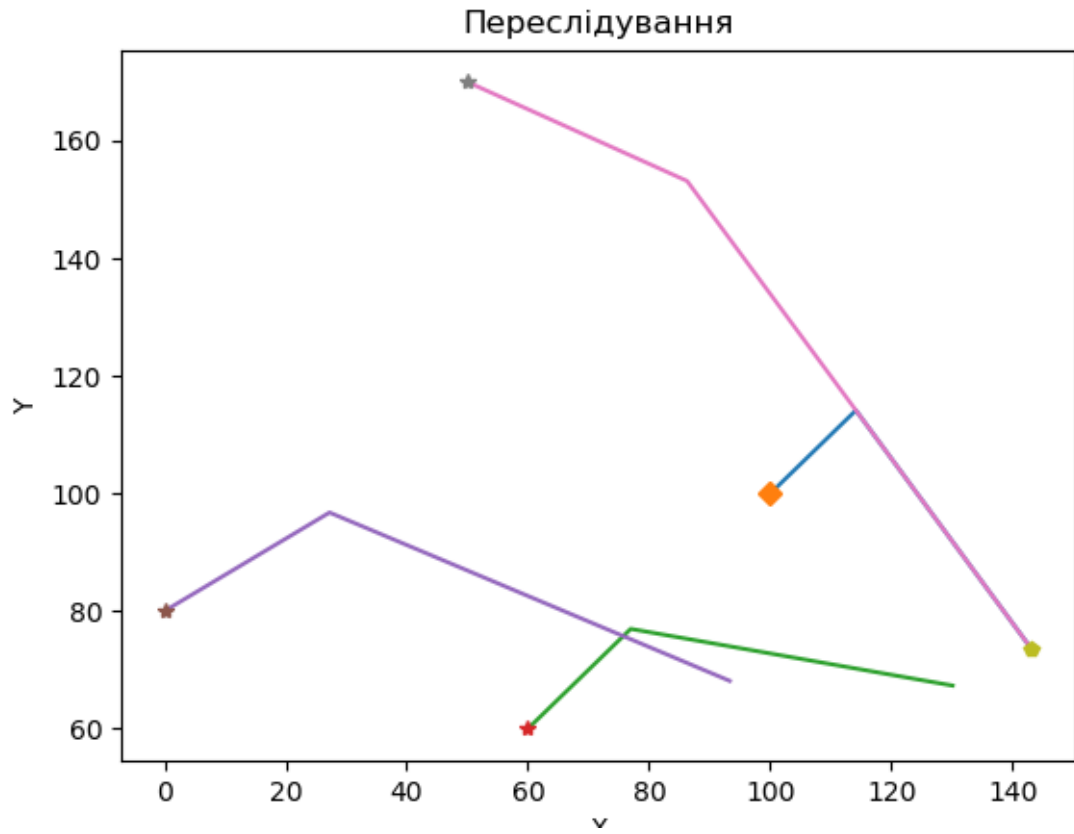


Рисунок 3.2 – 3 переслідувачі

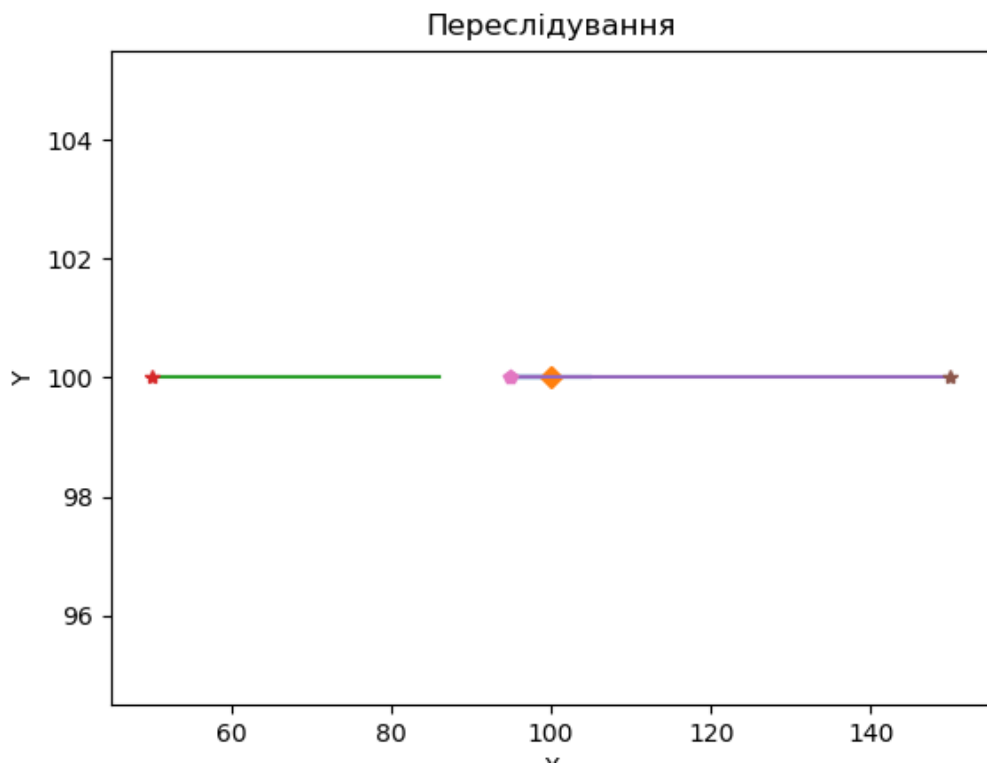


Рисунок 3.3 – Випадок спантеличення з 2 переслідувачами

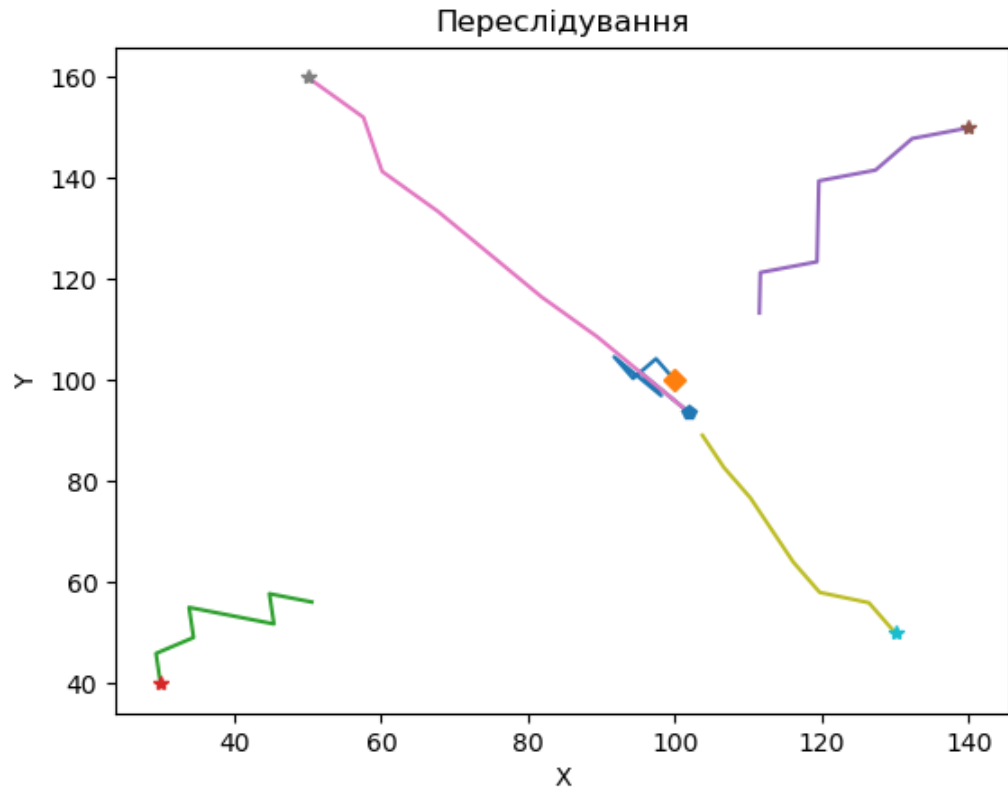


Рисунок 3.4 – Випадок спантеличення з 4 переслідувачами

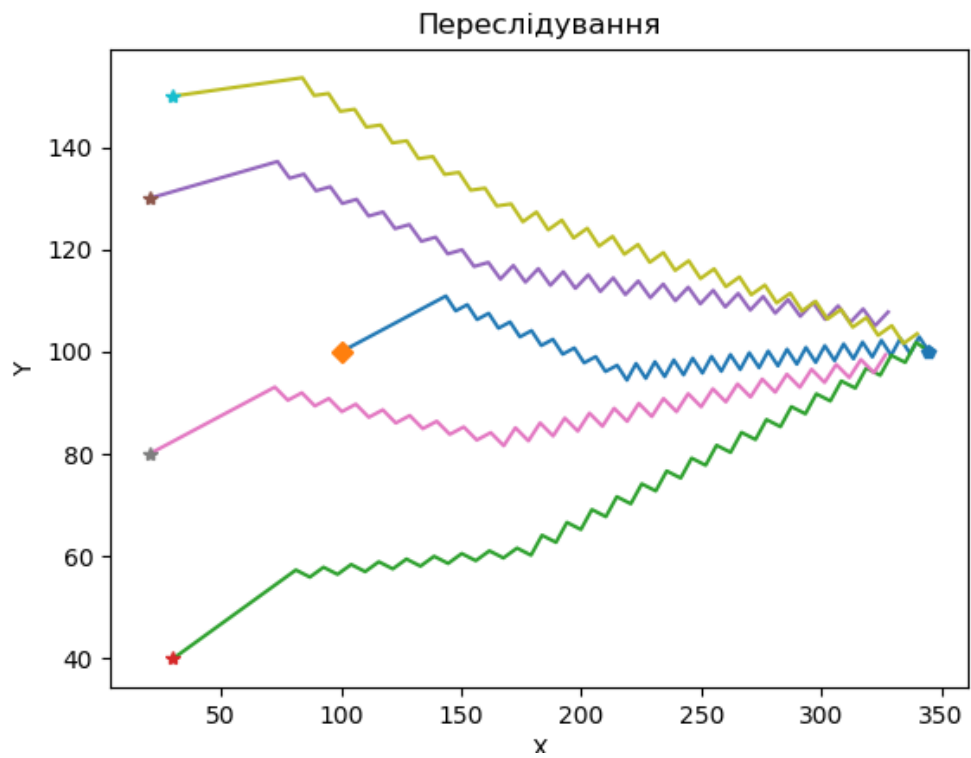


Рисунок 3.5 – 4 переслідувачі з однаковими швидкостями

На рис. 3.3 та рис. 3.4 зображенні випадки, коли вибір керування втікача виявився неефективним, оскільки він був в оточені.

Підсумовуючи моделювання даного випадку, можна сказати, що в задачі групового переслідування вибір керування втікача, якщо такий вибір існує, має опиратися на положення всіх переслідувачів та, можливо, інші відомості про їх рух.

У випадку №2 була здійснена взаємодія з користувачем, в майбутніх версіях ПП має бути можливість вводу керування через інтерфейс програми, але в цій версії ввід керування втікача здійснюється з консолі.

На рис. 3.6 - рис. 3.9 показані декілька прикладів при різній кількості переслідувачів та обмеженні швидкості.

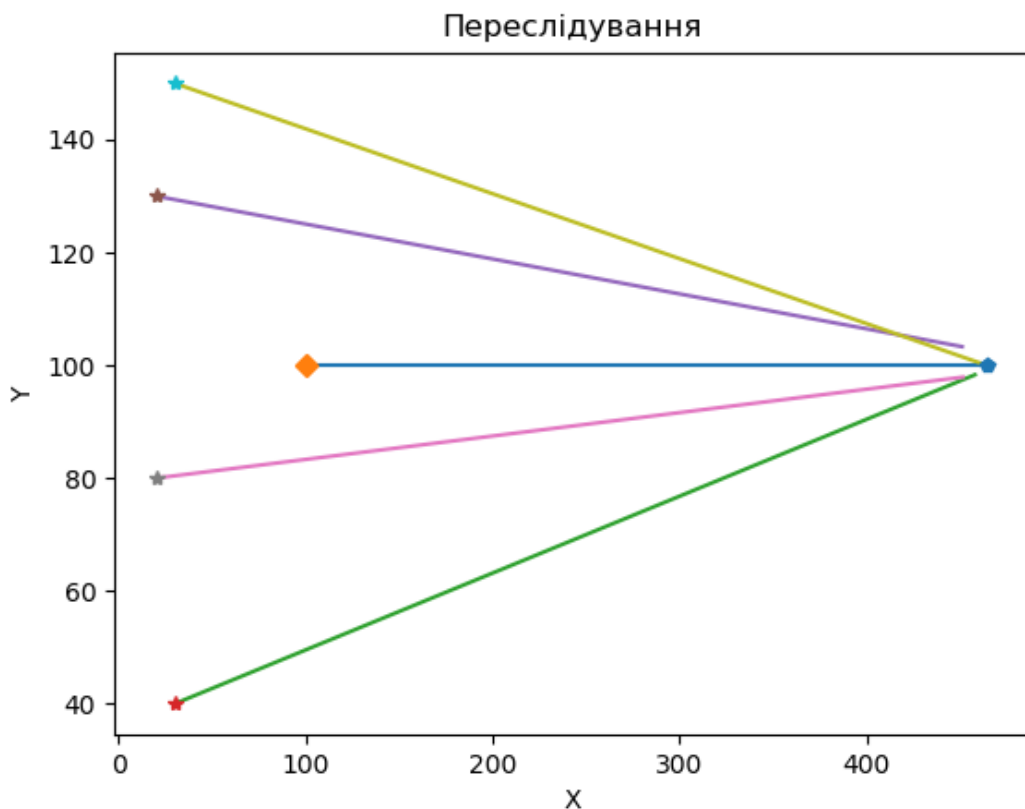


Рисунок 3.6 – 4 переслідувачі з одного боку та швидкостями

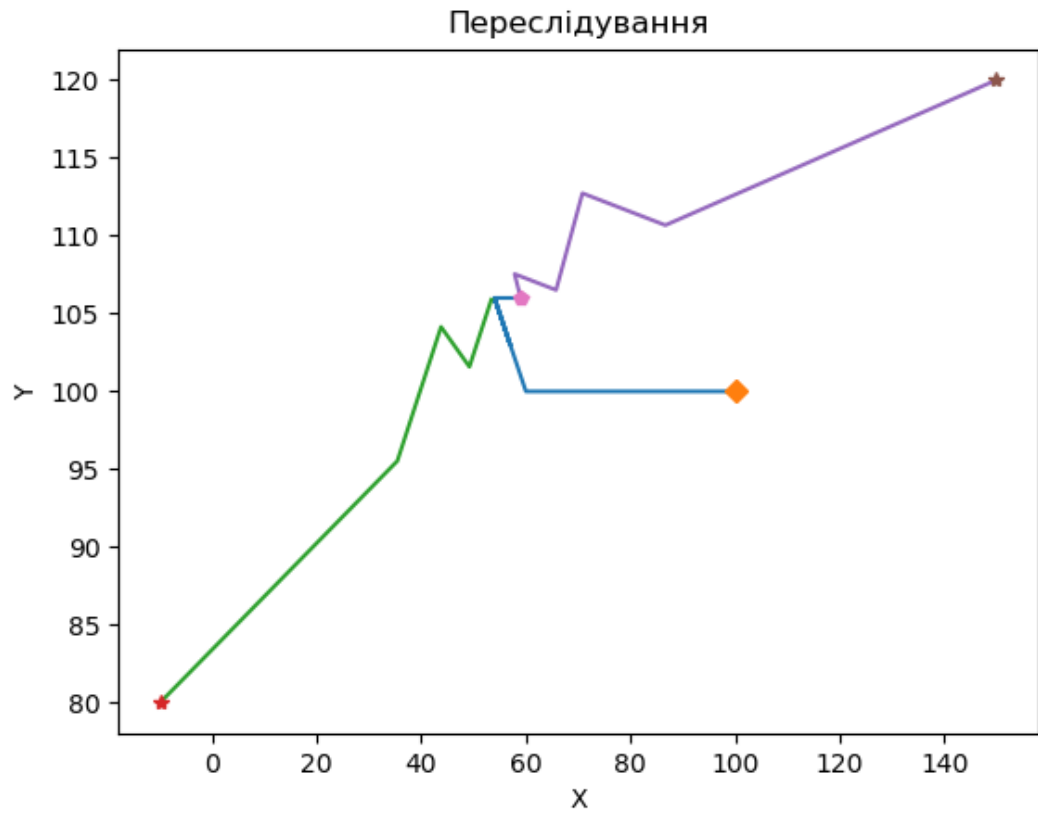


Рисунок 3.7 – 2 переслідувачі з оточенням

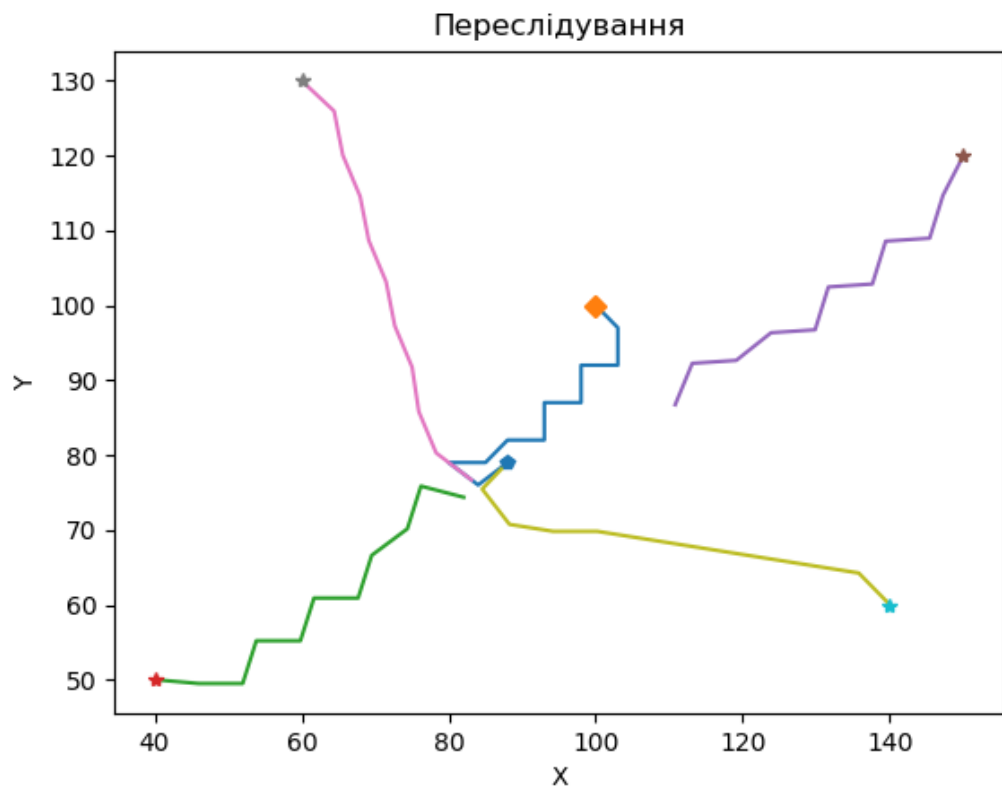


Рисунок 3.8 – 4 переслідувачі з оточенням

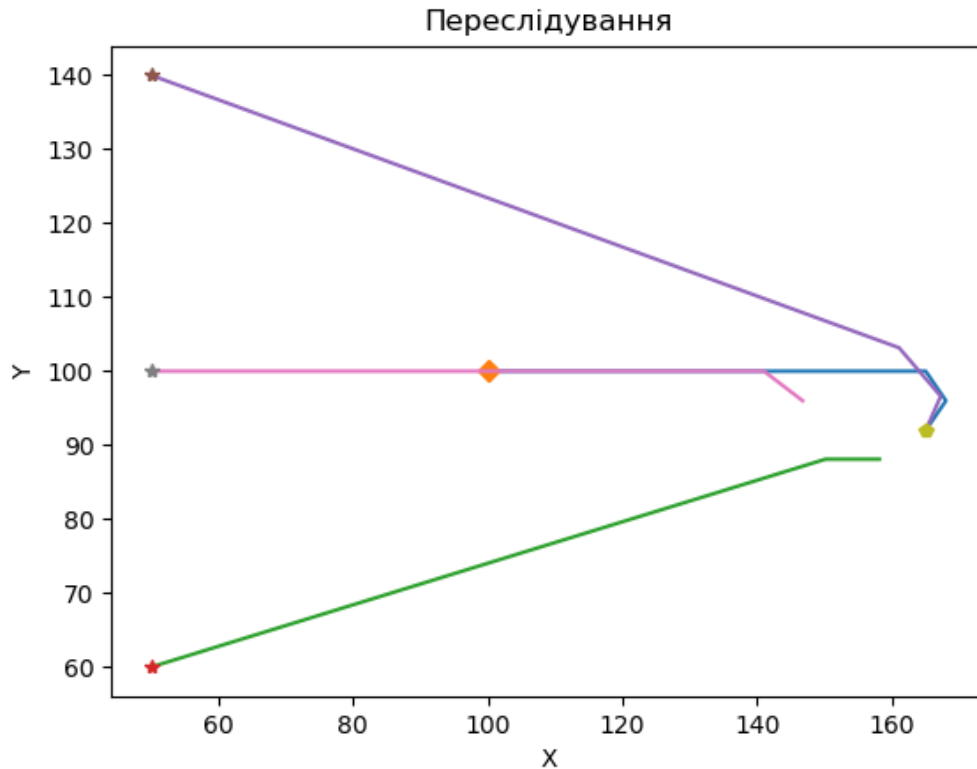


Рисунок 3.9 – 3 переслідувачі з однієї сторони

На рис. 3.6 можемо побачити, що втікачу рухатися вздовж прямої при тих умовах, що і на рис. 3.5, виявилося більш ефективним, оскільки його спіймали пізніше. На рис. 3.7, рис. 3.8 бачимо більш ефективну поведінку під час оточення.

Висновки до розділу 3

Був створений прототип системи проектування траєкторій в задачі групового переслідування з можливістю користувачу вибирати початкові умови задачі та керування втікача.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ ДЛЯ ВИРІШЕННЯ ГРУПОВОЇ ДИФЕРЕНЦІАЛЬНОЇ ГРИ ПЕРЕСЛІДУВАННЯ

4.1 Постановка задачі проектування

У даному розділі проводиться оцінка програмного продукту призначеного для вирішення диференційної групової гри переслідування. Програма була створена використовуючи мову програмування Python.

Середовищем розробки - редактор PyCharm, стандартне розширення .py дозволяє використовувати розроблений програмний продукт на будь-якій системі, незалежно апаратного забезпечення та її конфігурації.

Нижче наведено аналіз різних реалізацій гри переслідування з метою вибору оптимального, з огляду при цьому на основні фактори, які впливають на роботу програми та якість вихідного продукту.

4.2 Обґрунтування функцій програмного продукту

Основні функції продукту:

F_1 - вирішення задачі: а) вирішення конкретної задачі; б) вирішення певного класу задач; F_2 - алгоритм вирішення: а) використання відомого алгоритму; б) власна розробка.; F_3 - вибір мови програмування: а) мова програмування C++; б) мова програмування Python; в) мова програмування Java

Побудуємо морфологічну карту за даними варіантами (рис.4.1).

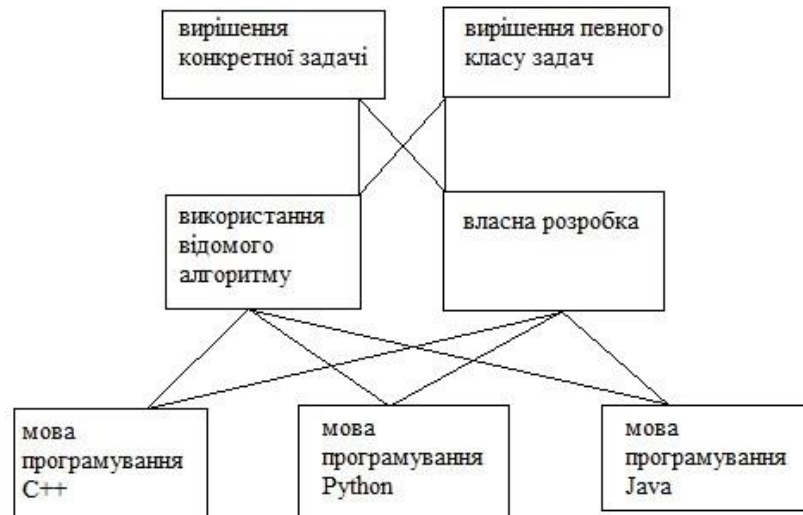


Рисунок 4.1 - Морфологічна карта

Позитивно-негативна матриця варіантів основних функцій зображена в табл. 4.1.

Таблиця 4.1 - Позитивно-негативна матриця

Основні функції	Варіанти реалізації	Переваги	Недоліки
F1	А	Швидкість розробки та спеціалізованість продукту	Складність при внесенні правок
	Б	Універсальність під заданий клас	Час на розробку
F2	А	Прогнозований результат роботи	Не унікальний продукт
	Б	Унікальність та швидкість роботи	Розробка власного алгоритму
F3	А	Швидкодія	Час на програмування
	Б	Функціонал	Час виконання
	В	Кросплатформеність, об'єктно-орієнтоване програмування	Час на програмування та виконання

4.3 Обґрунтування системи параметрів

Для оцінювання якості розглянутих реалізацій обрані параметри, описані нижче. Для визначення оцінки реалізацій виберемо параметри, описані в табл. 4.2. Також в цій таблиці наведені їх значення.

Таблиця 4.2 - Основні параметри програми

Назва параметру	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
час на оволодіння теорією	X1	год	100	50	20
час на вирішення задачі	X2	год	20	5	1
час на реалізацію програми	X3	год	60	30	10

За значеннями в таблиці побудуємо на рис. 4.2 - рис. 4.4 графічні характеристики параметрів

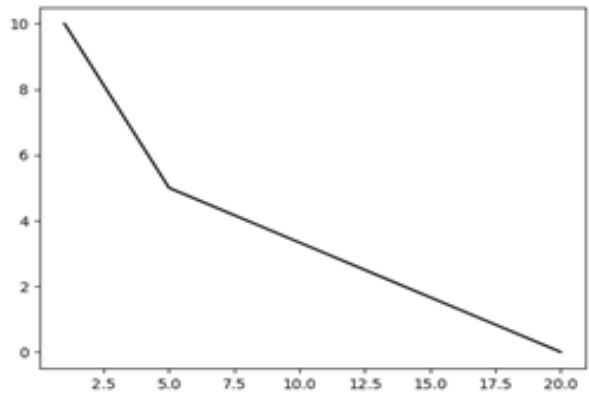
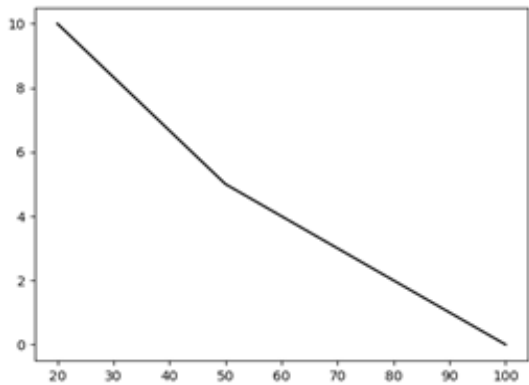


Рисунок 4.2 - Значення параметру X1 Рисунок 4.3 - Значення параметру X2

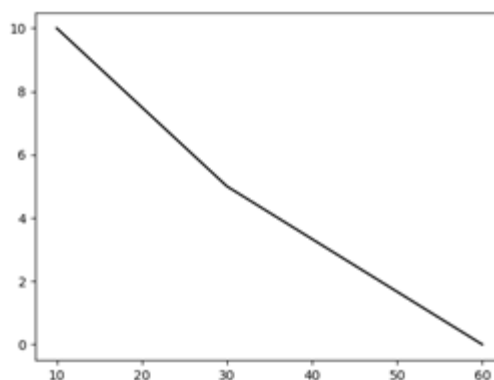


Рисунок 4.4 - Значення параметру ХЗ

Оцінемо вагомість параметрів методом попарного зрівняння. Ранги варіюються від 1 до 3. Найбільшим рангом приймаємо ранг 3,та, відповідно, найменшим є 1. Результати наведені в табл. 4.3- табл. 4.4

Таблиця 4.3 - Результат оцінки параметрів

Параметр	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	i^2
			1	2	3	4	5	6	7			
X1	час на оволодіння теорією	год	2	1	1	1	1	1	1	8	-6	6
X2	час на вирішення задачі	год	3	3	3	3	3	3	3	21	7	9
X3	Час на реалізацію програми	год	1	2	2	2	2	2	2	13	-1	
	Разом		6	6	6	6	6	6	6	42	0	86

Таблиця 4.4 - Попарне порівняння параметрів

Параметри	Експерти							Підсумкова оцінка	Коефіцієнт переваги (x_{ij})
	1	2	3	4	5	6	7		
x_1 та x_2	<	<	<	<	<	<	<	<	0.5
x_2 та x_3	>	>	>	>	>	>	>	>	1.5
x_3 та x_1	<	>	>	>	>	>	>	>	1.5

Коефіцієнт узгодженості дорівнює

$$W = \frac{12S}{N^2(n^3-n)} = \frac{12 \cdot 86}{49(27-33)} \approx 0.877 > 0.67$$

Отже, результати вважаємо достовірними. За результатами попарного порівняння обчислюємо вагомість кожного з критеріїв в табл. 4.5.

Таблиця 4.5 - Розрахунок вагомості параметрів

Параметри	Параметри			Перша ітерація		Друга ітерація	
	X1	X2	X3	b_i	K_{bi}	b_i^1	K_{bi}^1
X1	1	0.5	0.5	2	0.222	5.5	0.22
X2	1.5	1	1.5	4	0.444	11.5	0.46
X3	1.5	0.5	1	3	0.333	8	0.32
Всього:				9	1	25	1

4.4 Аналіз варіантів реалізації функцій

Проаналізувавши дані, ми можемо залишити лише такі варіанти:

$$F_1 \text{ Б} - F_2 \text{ А} - F_3 \text{ Б}$$

$$F_1 \text{ Б} - F_2 \text{ Б} - F_3 \text{ Б}$$

В табл. 4.6 наведене обчислення коефіцієнтів рівня якості.

Таблиця 4.6 - Обчислення коефіцієнтів рівня якості

Основні функції	Варіант реалізації функції	Параметри	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	Б	X1	40	6.67	0.22	1.47
		X2	5	5	0.46	2.3
F2	А	X1	30	8.34	0.22	1.84
		X2	4	6.25	0.46	2.88
	Б	X1	60	4	0.22	0.88
		X2	6	4.67	0.46	2.15
F3	Б	X3	20	7.5	0.32	2.4

Отже рівень якості для варіантів складе

$$K1 = 1.47 + 2.3 + 1.84 + 2.88 + 2.4 = 10.89$$

$$K2 = 1.47 + 2.3 + 0.88 + 2.15 + 2.4 = 9.2$$

З результатів впливає, що перший варіант є кращим.

4.5 Економічний аналіз варіантів розробки програми

Для визначення вартості розробки спочатку проведемо розрахунок трудомісткості. Обидва варіанти реалізації складаються з наступних завдань:

- 1) Розробка алгоритму.
- 2) Розробка програмного продукту.

Також кожен з варіантів має додаткове завдання. Для першого варіанту додаткове завдання:

- 3.1) Підготовка теоретичної бази для вирішення задачі.

Для другого варіанту додаткове завдання:

- 3.2) Створення власної теоретичної бази для вирішення задачі.

Для завдання 1: (алгоритм групи складності 1, ступінь новизни В, вид використаної інформації – НДІ) $TP = 43$ людино-днів, $K_n = 0.81$, $K_{ск} = 1$, $K_{стм} = 1$. Тому $T1 = 43 * 0.81 = 34.83$ людино дні.

Для завдання 2: (алгоритм групи складності 1, ступінь новизни А, вид використаної інформації – НДІ) $TP = 90$ людино-днів, $K_n = 1.7$, $K_{ск} = 1$, $K_{ст} = 0.7$. Тому $T2 = 90 * 1.7 * 0.7 = 107.1$ людино дні.

Для додаткового завдання 3.1: (алгоритм групи складності 3, ступінь новизни В, вид використаної інформації НДІ) $TP = 12$ людино-днів, $KП = 0.6$, $KСК = 1$, $KСТ = 0.7$, $KСТ.М = 1$. Тому $T31 = 12 * 0.6 = 7.2$ людино-днів.

Для додаткового завдання 3.2: (алгоритм групи складності 3, ступінь новизни А, вид використаної інформації НДІ) $TP = 27$ людино-днів, $KП = 1.26$, $KСК = 1$, $KСТ = 0.7$, $KСТ.М = 1$. Тому $T32 = 27 * 1.26 = 34.02$ людино-днів.

Звідси виходить, що повна трудомісткість першого варіанту

$$T = 34.83 + 107.1 + 7.2 = 149.13 \text{ людино-днів.}$$

А повна трудомісткість другого варіанту

$$T = 34.83 + 107.1 + 34.02 = 175.95 \text{ людино-днів.}$$

В розробці мають брати участь один програміст з окладом 14 000 грн та один математик з окладом 10 000. Погодинна заробітна плата працівників складе:

$$C = (14000 + 10000)/2 * 21 * 8 = 71.43 \text{ грн}$$

Поваріантно:

$$1) C = 149.13 * 71.43 * 1.2 * 8 = 102262.617 \text{ грн}$$

$$2) C = 175.95 * 71.43 * 1.2 * 8 = 120653.842 \text{ грн}$$

Відрахування ЄСВ складають 22%, або, поваріантно

$$1) C' = C1 * 0.22 = 102262.617 * 0.22 = 22497.78 \text{ грн}$$

$$2) C' = C2 * 0.22 = 120653.842 * 0.22 = 26543.85 \text{ грн.}$$

Визначаємо витрати на оплату однієї машино-години. З урахуванням заробітної плати програміста в розмірі 14000 грн з коефіцієнтом зайнятості 0.2, отримуємо:

$$C_{\Gamma} = 12 * M * K_3 = 12 * 14000 * 0.2 = 33600 \text{ грн.}$$

Включаючи додаткову плату:

$$C_{3\pi} = C_{\Gamma} * (1 + K_3) = 33600 * 1.2 = 40320 \text{ грн.}$$

Відповідно, відрахування на соціальний внесок складуть

$$C_{\text{від}} = C_{3\pi} * 0.22 = 40320 * 0.22 = 8870 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 10000 грн.

$$C_a = K_{TM} * K_a * C_{пр} = 1.15 * 0.25 * 10000 = 2875 \text{ грн.}$$

Профілактичні трати $C_p = K_{TM} * K_p * C_{пр} = 1.15 * 0.05 * 10000 = 575 \text{ грн.}$

Ефективний годинний фонд

$$T_{EF} = (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B = (365 - 104 - 11 - 16) \cdot 8 \cdot 0.9 = 1684.8$$

годин. Звідси виходить, що витрати на оплату електроенергії, з урахуванням ПДВ, складуть:

$$C_{EL} = 1684.8 * 0.2 * 1.46255 * 1.75 = 862.43 \text{ грн.}$$

Накладні витрати складуть $C_H = C_{пр} * 0.67 = 10000 * 0.67 = 6700 \text{ грн}$

Річні витрати на експлуатацію

$$C_{EKC} = C_{ЗП} + C_{ВІД} + C_A + C_P + C_{EL} + C_H = 40320 + 8870 + 2875 + 575 + 862.43 + 6700 = 60202.43 \text{ грн.}$$

Тоді собівартість однієї машино-години складе

$$C_{M-G} = C_{EKC} / T_{EF} = 60202.43 / 1684.8 = 35.73 \text{ грн/год.}$$

Тоді витрати на оплату машинного часу складають:

$$1) C_M = 35.73 * 149.13 = 5328.4149 \text{ грн.}$$

$$2) C_M = 35.73 * 175.95 = 6286.6935 \text{ грн.}$$

Накладні витрати становитимуть 67% від заробітної плати. Тобто, $0.67 * 102262.617 = 68515.54 \text{ грн}$ для першого варіанту та $0.67 * 120653.842 = 80838.0741 \text{ грн}$ для другого для другого варіанту

Таким чином, вартість розробки ПП та проведення дослідів складає:

Для першого:

$$102262.617 + 22497.78 + 5328.4149 + 68515.54 = 198604.352 \text{ грн.}$$

Для другого:

$$120653.842 + 26543.85 + 6286.6935 + 80838.0741 = 234322.46 \text{ грн.}$$

4.6 Вибір кращого варіанту ПП техніко-економічного рівня

Коефіцієнти техніко-економічного рівня, розраховані за формулою:

$$K_{\text{ТЕР}j} = K_{\text{К}j} / C_{\text{Ф}j}$$

$K_{\text{ТЕР}1} = 10.89 / 198604.352 = 5.48 \cdot 10^{-5}$ для першого варіанту реалізації.

$K_{\text{ТЕР}2} = 9.2 / 234322.46 = 3.93 \cdot 10^{-5}$ для другого варіанту реалізації.

Висновки до розділу 4

Отже, враховуючи всі дослідження, які ми провели, можна сказати, що перший варіант реалізації є найбільш оптимальним з розглянутої точки зору.

ВИСНОВКИ

В роботі був досліджений метод розв’язуючих функцій для певного класу задач групового переслідування. Сформульовано необхідні умови для завершення гри, наведено алгоритми побудови керування переслідувачами, проілюстровано схему методу розв’язуючих функцій на прикладі задачі для простого руху та прикладі задачі для процесу з простими матрицями.

Для представлення траєкторій рухів переслідувачів та втікача у грі з простим рухом та для випадку площини був створений програмний продукт на мові програмування Python. Програмний продукт є прототипом системи моделювання «переслідувачі-втікач», яку можна буде застосовувати для вибору керування в задачах керування. Для програми вхідними даними були наступні відомості: кількість переслідувачів; початкові координати всіх учасників; максимальні значення швидкості для всіх учасників. У розробці було розглянуто два випадки вибору керування втікача. Результати мають графічне представлення.

ПЕРЕЛІК ПОСИЛАНЬ

1. Айзекс Р. Дифференциальные игры. Москва: Мир, 1967. 480 с.
2. Альбрехт Э. Г. О сближении квазилинейных объектов в регулярном случае. Дифференциальные уравнения. 1971. Т.7. № 7 С. 1171-1178.
3. Батухтин В. Д. Экстремальное прицеливание в нелинейной игре. ДАН СССР. 1972. Т. 207. № 1. С. 11-14.
4. Ушаков В. Н. К задаче построения стабильных мостов в дифференциальной игре сближения-уклонения. Изв. АН СССР. Техническая кибернетика. 1980. № 4. С. 29-36.
5. Ледаев Ю. С., Мищенко Е. Ф. Об оптимальных стратегиях в дифференциальных играх фиксированной продолжительности. ДАН СССР. 1986. Т. 286. № 2. С. 284-287.
6. Левченков А. Ю. Об одной задаче сближения двух различных преследователей с одним убегающим. Прикладная математика и механика. 1998. Т. 52. Вып. 1, С. 3-8.
7. Чикрий А. А. Квазилинейные дифференциальные игры со многими участниками. ДАН СССР. 1979. Т.246. № 6. С. 1306-1309.
8. Понтрягин Л. С., Мищенко А. С. Решение линейной дифференциальной игры преследования на основе альтернированного интегрирования без дискриминации управления убегания. ДАН СССР. 1984. Т. 278. № 1. С. 1330-1334.
9. Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф. Математическая теория оптимальных процессов. Москва: Наука. 1969. 384 с.
10. Пшеничный Б. Н. Структура дифференциальных игр. ДАН СССР. 1969. Т. 184. № 2. С. 285-287.

11. Чикрий, А. А. Конфликтно управляемые процессы. Киев: Наук. думка, 1992. 384 с.

ДОДАТОК А ІЛЮСТРАТИВНІ МАТЕРІАЛИ ДО ДОПОВІДІ

ДИПЛОМНА РОБОТА НА ТЕМУ: «ГРУПОВА ДИФЕРЕНЦІАЛЬНА ГРА ПЕРЕСЛІДУВАННЯ»

ВИКОНАВЕЦЬ РОБОТИ:

СТУДЕНТ ІV КУРСУ,

ГРУПИ КА-63

ГИРЯВЕЦЬ ДМИТРО МИКОЛАЙОВИЧ

КЕРІВНИК:

КАНДИДАТ ФІЗ.-МАТ. НАУК,

ДОЦЕНТ

БАРАНОВСЬКА ЛЕСЯ ВАЛЕРІЙВНА

Київ-2020

ОБ'ЄКТ ДОСЛІДЖЕННЯ

- групова диференціальна гра переслідування.

МЕТА РОБОТИ

- побудувати керування за допомогою методу розв'язуючих функцій у груповій диференціальній грі переслідування.

МЕТОДИ ДОСЛІДЖЕННЯ

- побудува алгоритмів керування.

2

ЗАДАЧА ЗБЛИЖЕННЯ

$$\dot{z}_l = A_l z_l + \varphi_l(u_l, v), z_l \in R^{n_l}, l = \overline{1, \vartheta}, u_l \in U_l, v \in V$$

де $z_l = x_l - y$

Термінальна множина

$$M^* = \bigcup_{l=1, \dots, \vartheta} \{M_l^* + M_l\}$$

де M_l^* - лінійний підпростір з R^n , M_l - непорожні компакти з ортогональних доповнень L_l до M_l^* в просторі R^n .

3

БАГАТОЗНАЧНІ ВІДОБРАЖЕННЯ

Нехай π_l – оператор ортогонального проектування з R^n на L_l . Введемо багатозначні відображення

$$W_l(t, v) = \pi_l e^{A_l t} \varphi_l(U_l, v)$$

$$W_l(t) = \bigcap_{v \in V} W_l(t, v), t \geq 0, v \in V$$

$\gamma_l(t)$ – борелівський селектор відображення $W_l(t)$. Зафіксуємо його та покладемо

$$\xi_l(t, z_l, \gamma_l(\cdot)) = \pi_l e^{A_l t} z_l + \int_0^t \gamma_l(\tau) d\tau$$

4

РОЗВ'ЯЗУЮЧІ ФУНКЦІЇ

Розв'язуючі функції:

$$\begin{aligned} \alpha_l(t, \tau, z_l, v, \gamma_l(\cdot)) = \\ = \sup \{ \alpha_l \geq 0 : [W_l(t - \tau, v) - \gamma_l(t - \tau)] \bigcap \alpha_l (M_l - \xi_l(t, z_l, \gamma_l(\cdot))) \neq \emptyset \} \end{aligned}$$

Загальний час гри:

$$T_\vartheta(z, \gamma(\cdot)) = \min \{ t \geq 0 : \inf_{v(\cdot) \in \Omega_V} \max_{l=1, \dots, \vartheta} \int_0^t \alpha_l(t, \tau, z_l, v(\tau), \gamma_l(\cdot)) d\tau \geq 1 \}$$

Контрольна функція:

$$h_\vartheta(t) = h_\vartheta(T, t, z^\circ, v(\cdot), \gamma(\cdot)) = 1 - \max_{l=1, \dots, \vartheta} \int_0^t \alpha_l(T, \tau, z^\circ, v(\tau), \gamma_l^\circ(\cdot)) d\tau$$

5

АКТИВНЕ КЕРУВАННЯ

$$\begin{aligned} U_1^l(\tau, v) = \left\{ u_l \in U_l : \pi_l e^{A_l(T-\tau)} \varphi_l(u_l, v) - \gamma_l(T - \tau) \in \right. \\ \left. \alpha_l(T, \tau, z^\circ, v(\tau), \gamma_l^\circ(\cdot)) (M_l - \xi_l(T, z_l^\circ, \gamma_l^\circ(\cdot))) \right\} l = \overline{1, \vartheta} \end{aligned}$$

$$u_l(\tau) = u_1^l(\tau, v) = \text{lexmin } U_1^l(\tau, v), l = \overline{1, \vartheta}$$

На інтервалі $\tau \in [0, t_*]$

6

ПАСИВНЕ КЕРУВАННЯ

Для l_* , такого, що

$$1 - \int_0^t \alpha_{l_*}(T, \tau, z_{l_*}^\circ, v(\tau), \gamma_{l_*}^\circ(\cdot)) d\tau = 0$$

$$U_2^l(\tau, v) = \{u_l \in U_l: \pi_l e^{A_l(T-\tau)} \varphi_l(u_l, v) - \gamma_l^\circ(T-\tau) = 0\}$$

$$u_{l_*}(\tau) = u_2^l(\tau, v) = \text{lexmin } U_2^l(\tau, v)$$

а керування інших переслідувачів – довільні на інтервалі $(t_*, T]$

7

ПРОСТА ЗАДАЧА

Закони рухів переслідувачів мають вигляд

$$\dot{x}_l = u_l, \quad x_l \in R^k, k \geq 1, \|u_l\| \leq a_l, l = \overline{1, \vartheta}.$$

Закон руху втікача

$$\dot{y} = v, \quad y \in R^k, \|v\| \leq b.$$

8

АЛГОРИТМ ВИРІШЕННЯ

• Рахуємо $\alpha_l(z_l, v)$ за формулою

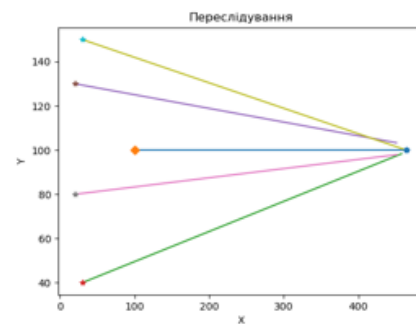
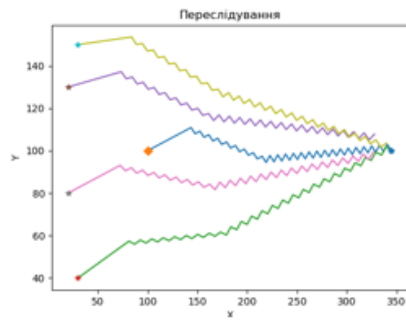
$$\alpha_l(z_l, v) = \frac{(z_l, v) + ((z_l, v)^2 + \|z_l\|^2(a_l^2 - \|v\|^2))^{1/2}}{\|z_l\|^2}, l = \overline{1, \vartheta}$$

• Перевіряємо, чи $\alpha_l(z_l, v) < 1, l = \overline{1, \vartheta}$. Якщо так, то керування рахуємо по формулам $u_l = v - \alpha_l(z_l, v)z_l, l = \overline{1, \vartheta}$

• Якщо для певних l : $\alpha_l(z_l, v) \geq 1$, то з них знаходимо переслідувача з найбільшим значенням l_{\max} : $\alpha_{l_{\max}} = \max \alpha_l(z_l, v)$ та визначаємо його керування $u_{l_{\max}} = v - z_{l_{\max}}$, а керування всіх інших покладаємо нулю.

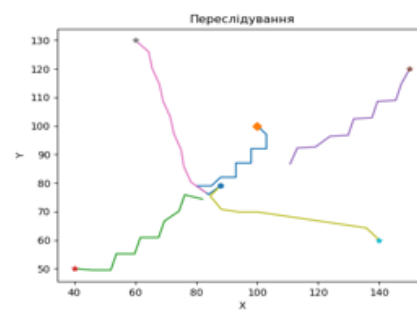
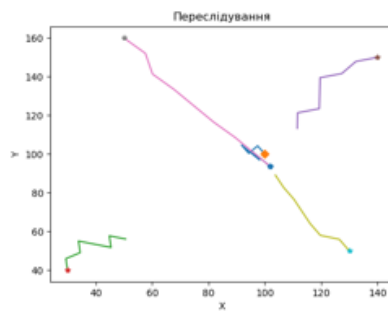
9

ВІЗУАЛІЗАЦІЯ РЕЗУЛЬТАТІВ



10

ВІЗУАЛІЗАЦІЯ РЕЗУЛЬТАТІВ



11

ВИСНОВКИ

- В роботі був досліджений метод розв'язуючих функцій для певного класу задач групового переслідування.
- Для представлення траєкторій рухів переслідувачів та втікача у грі з простим рухом та для випадку площини був створений програмний продукт на мові програмування Python.

12



ДЯКУЮ ЗА УВАГУ!

ДОДАТОК Б ТЕКСТ ПРОГРАМИ

1. Програмний код для випадку фіксованого керування втікача

```

import numpy as np
import pandas as pd
import math
import matplotlib.pyplot as plt
import matplotlib.animation as animation

def dist(a, b):
    k = 0
    for i in range(len(a)):
        k = k + (a[i] * b[i])
    return (k)

def norm(a):
    return math.sqrt(dist(a, a))

def find_z(a, b):
    z = []
    for i in range(len(a)):
        z.append([a[i][0] - b[0], a[i][1] - b[1]])
    return z

def sum_y(y, v):
    for i in range(len(y)):
        y[i] = y[i] + v[i]
    return 0

def find_v(z, v):
    distance = []
    for i in range(len(z)):
        distance.append([norm(z[i])])
    min_i = distance.index(min(distance))
    v_new = []
    for i in range(len(z[0])):
        v_new.append((-z[min_i][i]) * v / norm(z[min_i]))
    return v_new

def find_u_active(a, v, z):
    a_new = []

```

```

for i in range(len(z)):
    a_new_temp = []
    for j in range(len(z[0])):
        a_new_temp.append(v[j] - a[i]*z[i][j])
    a_new.append(a_new_temp)
return a_new

def find_a(z, v, b):
    a_new = []
    for i in range(len(b)):
        a_new.append(((dist(z[i], v)+math.sqrt(dist(z[i], v)**2 + norm(z[i])**2 *(b[i]*b[i] -
norm(v)**2)))/norm(z[i])**2)

    return a_new

def find_end(zi, vi):
    ok = []
    for i in range(len(zi)):
        ok.append(vi[i] - zi[i])
    return ok

handle = open("part1.txt")
dk = handle.read()
handle.close()
u_amount = int(dk.split()[0])
#v_amount = int(dk.split()[1])
x_0 = []
u_max = []
for iter in range(u_amount):
    x_0_temp=[]
    u_max.append((float(dk.split()[3+3*iter])))
    for jter in range(2):
        x_0_temp.append(float(dk.split()[1+3*iter +jter]))
    x_0.append(x_0_temp)
x_0 = np.array(x_0)
y_0 = []
for jter in range(2):
    y_0.append(float(dk.split()[1 + 3 * u_amount + jter]))
y_0 = np.array(y_0)

```



```

v_max = float(dk.split()[3 + 3 * u_amount])
"x_0 = np.array([[40., 60.],
    [-10., 80.],
    [150., 120.],
    [120., 170.],
    [200., 70.]])"
x = x_0
#u_max = [6, 8, 10, 12, 7]
#y_0 = np.array([100., 100.])
y = y_0
z_0 = find_z(x_0, y_0)
k = 0
point = 0
l=[]
X_mas=[]
Y_mas= []
X_mas.append(x.tolist())
Y_mas.append(y.tolist())
a_sum =0
while (k == 0):
    point = point +1
    z = find_z(x, y)
    v = find_v(z, v_max)
    a = find_a(z, v, u_max)
    if(max(a) < 1):
        u = find_u_active(a, v, z)
        x = x + u
        y = y + v
        #oj = sum_y(y, v)
    elif(max(a) == 1):
        u = find_u_active(a, v, z)
        x = x + u
        y = y + v
        #oj = sum_y(y,v)
    k = 1

```

```

else:
    max_i = a.index(max(a))
    u = [[0. for j in range(len(z[0]))] for i in range(len(z))]
    u[max_i] = find_end(z[max_i], v)
    x = x + u
    y = y + v
    #oj = sum_y(y, v)
    k = 1
    X_mas.append(x.tolist())
    Y_mas.append(y.tolist())
    print(a)
print("Summary steps:",point)
Y_mas = np.array(Y_mas)
X_mas = np.array(X_mas)
fig = plt.figure()
ax1 = fig.add_subplot(1, 1, 1)
xs1 = []
ys1 = []
xs = []
ys = []
for iter in range(len(u_max)):
    xs1.append(X_mas[0, iter, 0])
    ys1.append(X_mas[0, iter, 1])
for iter in range(len(u_max)):
    xs1[iter] = []
    ys1[iter] = []
def animate(i):
    ax1.clear()
    if (i < len(Y_mas)):
        xs.append(Y_mas[i, 0])
        ys.append(Y_mas[i, 1])
        for iter in range(len(u_max)):
            xs1[iter].append(X_mas[i, iter, 0])
            ys1[iter].append(X_mas[i, iter, 1])
    ax1.plot(xs, ys)

```

```

ax1.plot(y_0[0], y_0[1], marker='D')
for iter in range(len(u_max)):
    ax1.plot(xs1[iter], ys1[iter])
    ax1.plot(x_0[iter][0], x_0[iter][1], marker='*')
if (i >= len(Y_mas)):
    ax1.plot(xs[len(Y_mas)], ys[len(Y_mas)], marker='p')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Переслідування')
ani = animation.FuncAnimation(fig, animate, interval=50)
plt.show()

```

2. Програмний код для випадку взаємодії з користувачем

```

import numpy as np
import pandas as pd
import math
import matplotlib.pyplot as plt
import matplotlib.animation as animation

def dist(a, b):
    k = 0
    for i in range(len(a)):
        k = k + (a[i] * b[i])
    return (k)

def norm(a):
    return math.sqrt(dist(a, a))

def find_z(a, b):
    z = []
    for i in range(len(a)):
        z.append([a[i][0] - b[0], a[i][1] - b[1]])
    return z

def sum_y(y, v):
    for i in range(len(y)):
        y[i] = y[i] + v[i]
    return 0

def find_v(z, v):

```

```

distance = []
for i in range(len(z)):
    distance.append([norm(z[i])])
min_i = distance.index(min(distance))
v_new = []
for i in range(len(z[0])):
    v_new.append((-z[min_i][i]) * v / norm(z[min_i]))
return v_new

def find_u_active(a, v, z):
    a_new = []
    for i in range(len(z)):
        a_new_temp = []
        for j in range(len(z[0])):
            a_new_temp.append(v[j] - a[i] * z[i][j])
        a_new.append(a_new_temp)
    return a_new

def find_a(z, v, b):
    a_new = []
    for i in range(len(b)):
        a_new.append(
            (dist(z[i], v) + math.sqrt(dist(z[i], v) ** 2 + norm(z[i]) ** 2 * (b[i] * b[i] - norm(v) ** 2)))
            / norm(
                z[i] ** 2)
        )
    return a_new

def find_end(zi, vi):
    ok = []
    for i in range(len(zi)):
        ok.append(vi[i] - zi[i])
    return ok

fig = plt.figure()
ax1 = fig.add_subplot(1, 1, 1)
handle = open("part2.txt")
dk = handle.read()
handle.close()
u_amount = int(dk.split()[0])

```

```

#v_amount = int(dk.split()[1])
x_0 = []
u_max = []
for iter in range(u_amount):
    x_0_temp=[]
    u_max.append((float(dk.split()[3+3*iter])))
    for jter in range(2):
        x_0_temp.append(float(dk.split()[1+3*iter +jter]))
    x_0.append(x_0_temp)
x_0 = np.array(x_0)
y_0 = []
for jter in range(2):
    y_0.append(float(dk.split()[1 + 3 * u_amount + jter]))
y_0 = np.array(y_0)
v_max = float(dk.split()[3 + 3 * u_amount])
ax1.plot(y_0[0], y_0[1], marker='*')
for iter in range(len(u_max)):
    ax1.plot(x_0[iter][0], x_0[iter][1], marker='*')
'''
ax1.clear()
ax1.plot(y_0[0], y_0[1])
for iter in range(len(u_max)):
    ax1.plot(x_0[iter][0], x_0[iter][1])
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Переслідування')'''
'''x_0 = np.array([[40., 60.],
                  [70., 30.],
                  [150., 120.],
                  [50., 50.]])'''

x = []
x = x_0
'''u_max = [6, 8, 10, 7]
v_max = 5
y_0 = np.array([100., 100.]'''

```

```

y = y_0
z_0 = find_z(x_0, y_0)
control = True
point = 0
l = []
handle = open("Diploma.txt", 'w')
for ui in range(len(x)):
    for uo in range(len(x[ui])):
        handle.write('%s' % str(x[ui][uo]) + ' ')
for ui in range(len(y)):
    handle.write('%s' % str(y[ui]) + ' ')
handle.write("\n")
handle.close()
def animate(i):
    global control
    global point
    point = point + 1
    if(control != False):
        handle = open("Diploma.txt")
        dd = handle.read()
        handle.close()
        x = []
        y = []
        v = []
        ol = (len(u_max) + 1) * len(x_0[0])
        amount = int(len(dd.split()) / ((len(u_max) + 1) * len(x_0[0])))
        for ui in range(len(u_max)):
            x_new_temp = []
            for oi in range(len(x_0[0])):
                x_new_temp.append(float(dd.split()[ol * (amount - 1) + ui * len(x_0[0]) + oi]))
            x.append(x_new_temp)
        for ui in range(len(x_0[0])):
            y.append(float(dd.split()[ol * (amount - 1) + len(u_max) * len(x_0[0]) + ui]))
        z = find_z(x, y)
        f = True

```

```

print('pursuers:')
for h in range(len(x)):
    print((h+1), ' pursuer:', x[h])
print('You:', y)
while (f == True):
    print('input velocity')
    try:
        num = map(float, input().split())
        v = list(num)
        print(v)
    except (TypeError, ValueError, IndexError, UnboundLocalError):
        print('Try again')
    if (len(v) != 2):
        print('Input two numbers')
    elif(v[:] is str):
        print('Input two numbers')
    elif (norm(v) > v_max):
        print('Wrong. your maximum', v_max)
    else:
        f = False
a = find_a(z, v, u_max)
if (max(a) < 1):
    u = np.array(find_u_active(a, v, z))
    x = x + u
    sum_y(y, v)
elif (max(a) == 1):
    u = np.array(find_u_active(a, v, z))
    x = x + u
    sum_y(y, v)
    control = False
    print("You lost")
else:
    max_i = a.index(max(a))
    u = [[0. for okr in range(len(z[0]))] for ojk in range(len(z))]
    u[max_i] = find_end(z[max_i], v)

```

```

    u = np.array(u)
    x = x + u
    sum_y(y, v)
    control = False
    print("You lost")
print("Summary steps:", point)
handle = open("Diploma.txt", 'a')
for ui in range(len(x)):
    for uo in range(len(x[ui])):
        handle.write('%s' % str(x[ui][uo]) + ' ')
for ui in range(len(y)):
    handle.write('%s' % str(y[ui]) + ' ')
handle.write("\n")
handle.close()
X_mas = []
Y_mas = []
ol = (len(u_max) + 1) * len(x_0[0])
handle = open("Diploma.txt")
dk = handle.read()
handle.close()
amount = int(len(dk.split()) / ((len(u_max) + 1) * len(x_0[0])))
for j in range(amount):
    x_new = []
    y_new = []
    for ui in range(len(u_max)):
        x_new_temp = []
        for oi in range(len(x_0[0])):
            x_new_temp.append(float(dk.split()[ol * j + ui * len(x_0[0]) + oi]))
        x_new.append(x_new_temp)
    for ui in range(len(x_0[0])):
        y_new.append(float(dk.split()[ol * j + len(u_max) * len(x_0[0]) + ui]))
    X_mas.append(x_new)
    Y_mas.append(y_new)
X_mas = np.array(X_mas)
Y_mas = np.array(Y_mas)

```



```

xs1 = []
ys1 = []
xs = []
ys = []
for iter in range(len(u_max)):
    xs1.append(X_mas[0, iter, 0])
    ys1.append(X_mas[0, iter, 1])
for iter in range(len(u_max)):
    xs1[iter] = []
    ys1[iter] = []
for p in range(amount):
    xs.append(Y_mas[p, 0])
    ys.append(Y_mas[p, 1])
for iter in range(len(u_max)):
    for p in range(amount):
        xs1[iter].append(X_mas[p, iter, 0])
        ys1[iter].append(X_mas[p, iter, 1])
ax1.clear()
ax1.plot(xs, ys)
ax1.plot(y_0[0], y_0[1], marker='D')
for iter in range(len(u_max)):
    ax1.plot(xs1[iter], ys1[iter])
    ax1.plot(x_0[iter][0], x_0[iter][1], marker='*')
if (control==False):
    ax1.plot(xs[len(xs)-1], ys[len(ys)-1], marker='p')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Переслідування')
ani = animation.FuncAnimation(fig, animate, interval=1000)
plt.show()

```